INSIDE
AEGIS ANIMATOR™

A users guide to mastering Animator on the Amiga.™

by John A. S. Skeel

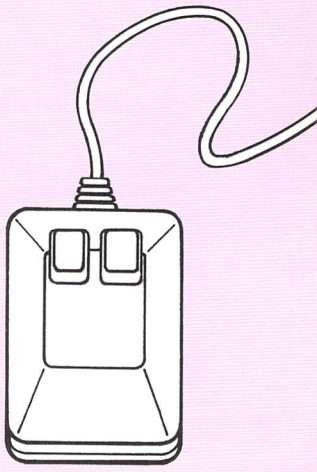
SER. NO. 1002168

Inside Aegis Animator

by

John A. S. Skeel

Published by
PaperDisk Publishing
A Division of Aegis Development, Inc.
Los Angeles, California



Inside Aegis Animator

Copyright © 1986 by PaperDisk Publishing.

All rights reserved. Printed in the United States of America. No part of this book may be reproduced in any form or by any means, or stored in a data base or retrieval system, without the express prior written permission of the publisher except in the case of brief quotations embodied in critical articles and reviews.

ISBN no. 0-915595-10-9

Trademarks and Copyrights

Amiga is a trademark of Commodore Business Machines, Inc.
GenLock, Amiga 1300 Genlock System is a trademark of Commodore Business Machines, Inc.
Aegis Animator and Aegis Images is a trademark of The Next Frontier Corporation.

Acknowledgments

This book, like any other, is the product of many people. Their ideas and hard work combined to make it happen, and it is with the greatest thanks that I dedicate this book to:

Jim Kent and all the folks at Island Graphics for creating the Animator program.

Tom Nelson for his assistance in the art work.

Paul Smith of DynaType for his editing and typesetting expertise.

Peter Green and Chris Savage of Peter Green Design for the cover design and artwork.

Alfredo Lavadenz of AVL Press for his production skills.

The men and women of Commodore-Amiga for producing a truly remarkable machine.

And most of all to Nancy Kildow for correcting my grammar and being a wellspring of support and love.

—J.A.S.S.

CONTENTS

Inside Aegis Animator

Chapter One:

Introduction 1
What You Can Do With Animator 2
About This Book 3

Chapter Two:

Getting Started 5
Loading Animator 6
A Quick Tour 7
The Fast Menu 11

Chapter Three:

How Animator Works 13
About Tweening 14
Metamorphic Animation 14
A Quick Animation 15

Chapter Four:

Making Shapes 23
Lines 24
Polygons 25
Circles 25
Stars 26
Blocks 26
Clone and Destroy 27
Clone 27
Destory 28

Chapter Five:

Movement 29
Sideways 30
In and Out 31
In Plane 33
Around X 34
Around Y 35
Rotation 32
Size 37
Path 38
Change Color 39
Change Type 40
Morph 41
Combinations 42

Chapter Six:

Select 45
Point 47
Points 47
Segment 48
Polygon 48
Polygons 49
All 49

Chapter Seven:

Time and Color 51
Tweens and Time 52
Color Control 55
A Note About the Color Bar 58
Cycle 58
Fade 59
Wipe 60
Range 60
Spectrum 61
Saving and Loading a Palette 61

Chapter Eight:

The Storyboard 63
Speed 65
Status 65
Exit 65
Edit 65
Splice 66
Cut 66
Delete 67
Activate 67
Storyboard and Paint Program Images 68
Memory Panic—System Going Down 68

Chapter Nine:

Using Images Created Outside Animator 69
Backdrops 70
Windows 71
Masks 73

Chapter Ten:

Storage 75

Chapter Eleven:

Animation Technique 79
Planning 81
Speed 82
Upstream 83
Script Files 83
Shortcuts for Loading Animator 85
Using Animator with Genlock 86

Chapter Twelve:

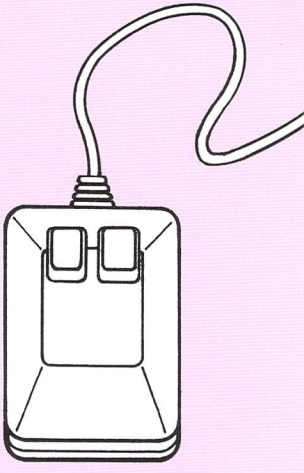
Aegis Animator Reference 87
Glossary 94
Index 98

Figures:

Figure 2.1 — The Main Screen 7
Figure 2.2 — The Storyboard 9
Figure 3.1 — Creating a Polygon 16
Figure 3.2 — Moving a Polygon Sideways 17
Figure 3.3 — Rotating a Polygon in the Same Plane 18
Figure 3.4 — Rotating a Polygon Around an X Axis 19
Figure 3.5 — Rotating a Polygon Around a Y Axis 20
Figure 3.6 — Changing the Size of a Polygon 21
Figure 4.1 — Cloning a Polygon 28
Figure 5.1 — Moving an Object Within the Same Plane 31
Figure 5.2 — Rotating a Polygon in Plane 33
Figure 5.3 — Rotating a Polygon Around an X Axis 35
Figure 5.4 — Rotating a Polygon Around a Y Axis 36
Figure 5.5 — Changing the Size of a Polygon 37
Figure 7.1 — An Animation of a Bouncing Ball 53
Figure 7.2 — The Time Requestor 54
Figure 7.3 — The Color Requestor 56
Figure 10.1 — The Storage Requestor 77
Figure 12.1 — The Fast Menu 93

CHAPTER ONE

Introduction



Welcome to Aegis Animator, the color animation tool for the Amiga. Whether you plan to use Animator for fun or business you'll find it an easy and effective tool.

With your Amiga and Aegis Animator you can:

- produce objects in 32 different colors from the full range of 4096 available.
- animate those objects by changing their shape, size, or color, or by dragging, rotating, or plotting a path for them about the screen.
- create up to nine different scripts and cut, or splice them together on a single storyboard.
- load backdrops from Aegis Images™ or similar paint packages.
- have a lot of fun.

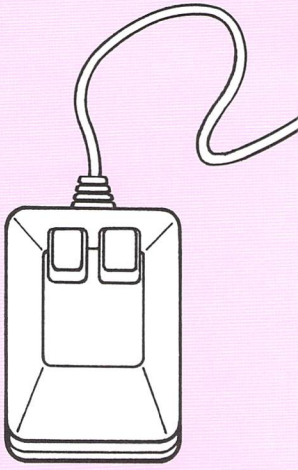
To use Aegis Animator you need to know how to use disks, the mouse, and other features of the Amiga. The basics are described in a manual supplied by Commodore entitled *Introduction to Amiga*. Take time to read this introduction now if you are not familiar with the workings of the machine. It will make learning Animator faster and more enjoyable.

About This Book

This book is laid out so the reader can use it beginning to end or skip about. Chapters 2 and 3 will give you the basics on how to use Animator. Each succeeding chapter will detail a different tool set of the program. Chapter 11 discusses the "how to" of animation and Chapter 12 is a quick reference organized by menu. Each menu item has a brief description and references to more detailed information.

CHAPTER TWO

Getting Started



First, turn on your Amiga and open the Workbench or, if you prefer, the CLI (Command Line Interpreter). (To learn about the Workbench and how to open tools, see *Introduction to Amiga*.).

The first thing you will want to do is make a working copy of your Animator disk. This is to protect you in case disaster should befall and your disk becomes inoperable. (To learn how to copy disks see *Introduction to Amiga*). Once you have a working copy, put the original away in a safe place and load the working copy into the disk drive.

If you are using the Workbench:

Open the Animator disk by clicking the mouse cursor on the disk icon. In the window that appears you'll see a tool icon labeled **Animator**. To open Aegis Animator, select the icon and choose **Open** from the **Project** menu.

If you are using the CLI:

Type **Ani** at the 1> prompt and press return.

In a few moments you'll see the Animator screen. The menu bar at the top shows you are ready to create polygons. Pressing the Menu (right) mouse button will display the main menu for Animator.

To choose any function from the menu, hold down the right mouse button and drag the cursor until it is over a menu selection. A sub-menu box will appear below it. You may choose any of these options by dragging the cursor until it highlights the item of your choice and then releasing the button.

A Quick Tour

Besides the main menu bar at the top, the main screen of Animator has a window in the upper left corner. This window is the fast menu. You can put it away at any time by using the close box. The close box is the small square in the upper left hand corner of the window. By clicking the mouse cursor (with the left button) in the box, the window will close and be removed from the screen. To bring the fast menu back at any time, select **fast menu** from beneath the **Project** menu choice.

The fast menu is an icon driven version of the main menu bar. It allows you to perform the most often used tasks quickly.

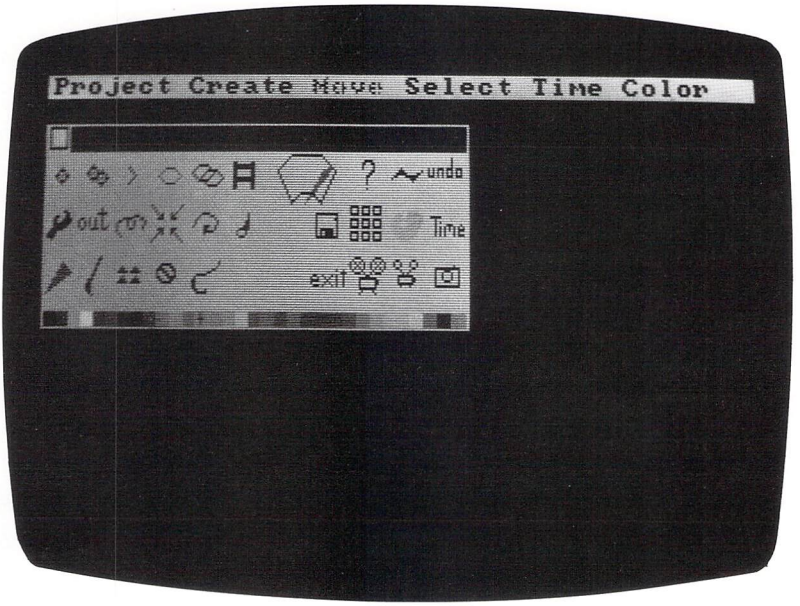


Figure 2.1 — The Main Screen.

Pull down the **Project** menu (hold the right button down with the mouse cursor over the **Project** selection). You'll see that the sub-menu box is broken into three sections.

The first two are options you would use most often. They are **undo** and **fast menu**. The latter as decribed above displays or removes the fast menu, and **undo** un-does the last action you performed. It's your "oops, I didn't mean to do that" selection.

The next group of selections perform tasks such as starting over (**new script**), loading saved scripts or other files (**storage**), splicing animations together (**storyboard**), control of speed and color (**time**, **color**), and displaying the amount of available memory (**status**). The last selection on the menu, **exit**, will exit animator and put you back in the Workbench (or CLI).

To the right of **Project** on the menu bar is **Create**. The sub-menu block beneath it contains the choices for creating the different shapes such as circles, stars, and multi-sided polygons you will use to produce the animations. At the bottom, are two selections that allow you to make exact copies (**clone**), or to remove a creation from the animation (**destroy**). If you drag the cursor down until the highlight bar is over **polygon**, you will see that another sub-menu box opens to the right. In this case it contains two choices **filled** and **outline**. By moving the cursor over to this box and releasing the mouse button when the highlight bar is on one of these choices you can choose between creating a filled polygon or just the outline of the shape you build.

The next choice to the right is **Move**. This group controls the different methods of moving an object about the screen. These include dragging, rotating, changing size, plotting a path, changing the color or type, and **morphing** (changing the actual physical characteristics of an object).

One more menu item to the right brings you to **Select**. This menu choice controls what items or portions of items on the screen you want to affect. For example, from the **Move** menu you may choose **size**. The **Select** menu determines which items on the screen will change size. You can choose a single (**polygon**)

or multiple (**polygons**) objects, or just a portion (**segment**) of an object.

The next selection is **Time**. The choices under **Time** perform functions like moving to the next tween (more on this later), replaying the animation created in the current tween or in all tweens, and playing the animation in a continuous loop. The last two items are preferences. The first, **ghost mode**, displays the entire animation in outline form only. This is useful for seeing objects that are currently behind other objects on the screen. The other choice, **see tween**, allows you to choose whether you view a tween as it exists at its beginning or as it is after it finishes (more later, I promise).

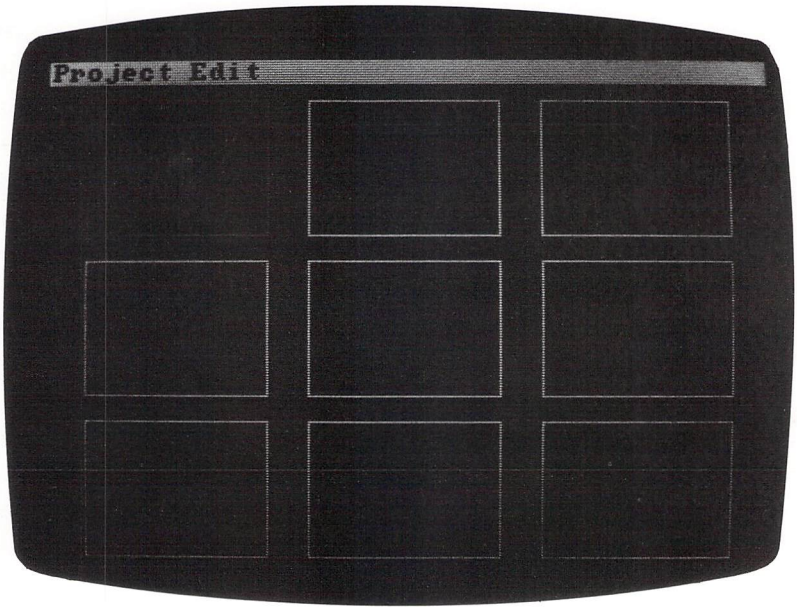


Figure 2.2 — The Storyboard.

The last choice on the bar is **Color**. If you have used your Amiga very much you have probably seen this type of menu already. It allows you to choose the color you will be working in when you create the different objects on the screen. To choose

a color, drag the cursor around the sub-menu box until the highlight box is on the color you want and then release the mouse button. If the fast menu is on the screen you will note that the palette icon will change to the new color you chose.

There is one more place to visit before we end the tour. Pull down the **Project** menu and choose **storyboard**. The screen will change to display nine windows and the menu bar will read **storyboard**.

The first window should be highlighted. This is the animation we just came from. It should be empty like the others unless of course you made something. These are the nine animations you can have in Animator at the same time (providing you have enough memory). From here you can cut and splice animations together. This allows you to create different pieces of an animation together and then choose the best ones to piece together in a final version.

Note that the cursor is in the shape of the word **into**. This is the function you are currently using. To enter any animation or a blank script, point to the one you want and click the Selection (left) mouse button. To return here choose **storyboard** from beneath the **Project** menu again.

If you press the Menu (right) mouse button you will see only two menu selections in this portion of the program. The first, **Project**, contains functions that allow you to change the speed and check on available memory. There is also a selection (**about ani**) that gives a history of the program, and another version of the **exit** function. It too will exit you from Animator and return you to the Workbench (or CLI).

The other storyboard menu choice is **Edit**. This is the option that allows you to **cut**, **splice**, and **delete** animations. The **activate** option sets the animations in motion so you can watch the different scripts playing side by side.

To get back to the main part of Animator, place the cursor (it should read **into**) on the first window and click the Selection button.

The Fast Menu

As mentioned above, the fast menu is a short cut method of choosing tools without having to pull down the menus. To the first time user it's a little confusing since most of the tools are unfamiliar. As you use Animator more and become familiar with the different icon shapes, though, the fast menu becomes easy to read.

The most important part of the fast menu to the beginner is the question mark in the upper right hand corner. This is the fast menu help function. By clicking the Selection button while the cursor is on this icon you activate it. The cursor will change into a question mark. Move the question mark to any item in the fast menu and press the Selection button again. The menu bar at the top of the screen will display a one line description of the function. For example, select the question mark and then select the camera icon in the lower right corner of the fast menu window. The bar at the top of the screen should now read **advance to next tween**. Selecting this icon is the same as choosing **next tween** from beneath the **Time** menu selection.

When a tool is in use or an option selected it will be highlighted in the fast menu. The palette icon, which is used for accessing the color control requestor, always displays the current color. To change the color, use the cursor to select a color from the color bar at the bottom of the fast menu window. The palette will change to that color.

The fast menu is a standard Amiga window so you may move it anywhere on the screen. To do this, place the cursor in the title bar at the top of the window and hold down the Selection button. The window will change to an outline and as long as you hold the button down you may move the outline to any part of the screen. Release the button when the window is where you want it and the the fast menu will reappear. When you are using tools to create or move objects on the screen, the fast menu will temporarily disappear to give you a clear field of view.

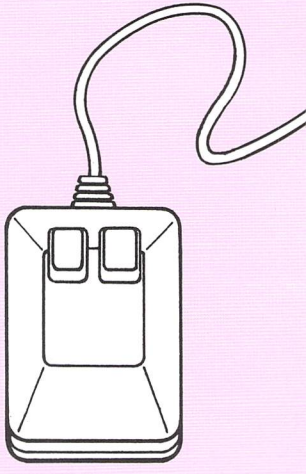
To remove the fast menu from the screen, select the close

box in the extreme upper left corner of the window. Chapter 12 has a detailed description of the fast menu.

Well, that completes the fifty cent tour. Now, it's time to get down to business and create some animations. This is covered in the next chapter.

CHAPTER THREE

How Animator Works



Aegis Animator is a metamorphic animation package that uses the concept of tweening to control movement. What that means to you is simply this. You create an image and then change its position or shape on the screen in a segment of time. Each segment of time is a tween. So you're concerned with what the shape will look like at the beginning of a tween and what you want it to look like at the end of the tween. Animator will handle the gradual or not so gradual change from the first shape into the next.

This concept differs from the better known cel animation method. In cel animation, to create the effect of a bird flying, you would first draw a bird with wings spread, let's say in the up position. The wings would be made on one piece of cellophane (hence, cel) that is laid over the birds body painted on another piece.

That same image is then reused with the wings redrawn a fraction further down creating the next image (sometimes called a frame). And then used again, this time drawing the wings yet a little further down. This would be the next frame. This process is repeated until the wings are all the way into the down position. The same images are then used in reverse moving the wings into an up position. At the same time you would want to move the bird (both pieces of cellophane) forward in relation to the

background to give the bird the effect of forward motion. When these images are played back in rapid succession the bird would seem to flap its wings and fly forward.

With Animator, the bird could be made to fly by changing the shape of the wing, say from its up position to a position in the middle. This would occur in the first tween. Then, the wing's shape would be changed from its middle position to its down position in the next tween. At the same time the bird could be pulled forward a little in each tween to create the forward motion. When the two tweens are replayed the wing will shift from its up position to its down position in one smooth movement and the bird will move ahead.

We'll discuss tweens in more detail later, but first let's run through a quick animation to see how everything fits in.

A Quick Animation

First, be sure you have a clean script to begin with.

- Select **new script** from the **Project** menu.

  A requestor will appear asking if you want to abandon the current script you are working on. Select **yes**.

- Close the fast menu by selecting the close box.

  During this example we'll just use the menus.

- Select **polygon** and then **filled** from the **Create** menu.

  When **polygon** is highlighted, move the cursor to the right until **filled** is also highlighted. Release the mouse button. The cursor will change into the polygon create tool.

- Place the cursor near the center of the screen and press the Selection (left) button.

  As you move the mouse, you will see a line appearing from the original point to the cursor. This is the first side of your polygon.

- Move the cursor to the right and then click the selection button again. Now move the cursor toward the bottom of the screen. A triangle will appear. Click the button again.

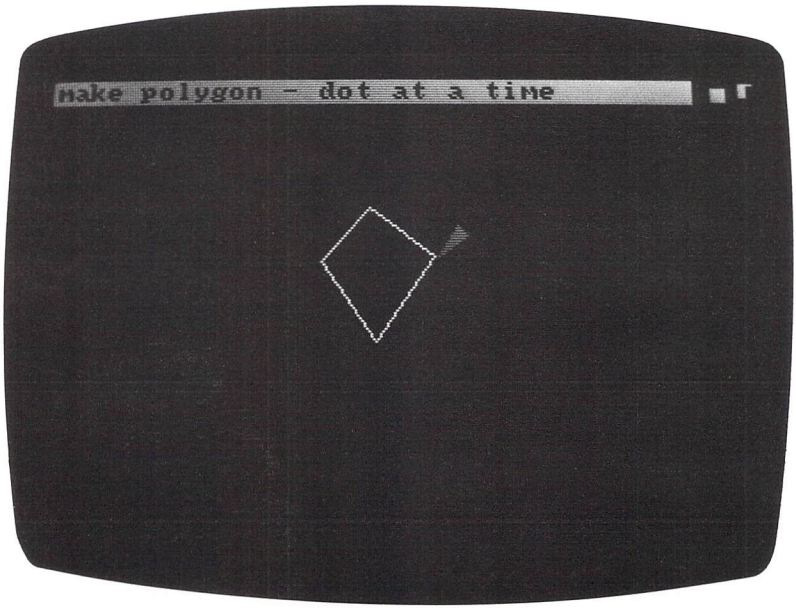


Figure 3.1 — Creating a Polygon

You now have two firm sides to your polygon and two rubber band sides.

- Pull the cursor toward the lower left of the screen until you have a kite-like shape. Click the Selection button again. Now to complete the shape click the Menu (right) mouse button.

  The object will be filled with the current color selected under the **Color** menu. You should have a kite flying at an angle on the screen. Now we'll make it fly.

- Select **move** and then **sideways** from the **Move** menu.

The cursor will change into a hand. The tip of the hand is the **hot** spot of the cursor. The hot spot is where the program looks when you click a mouse button to issue a command. For

example, if you had several small objects very close together and decided to move one of them, Animator would move the one under the tip even though other parts of the hand may be covering some of the other objects.

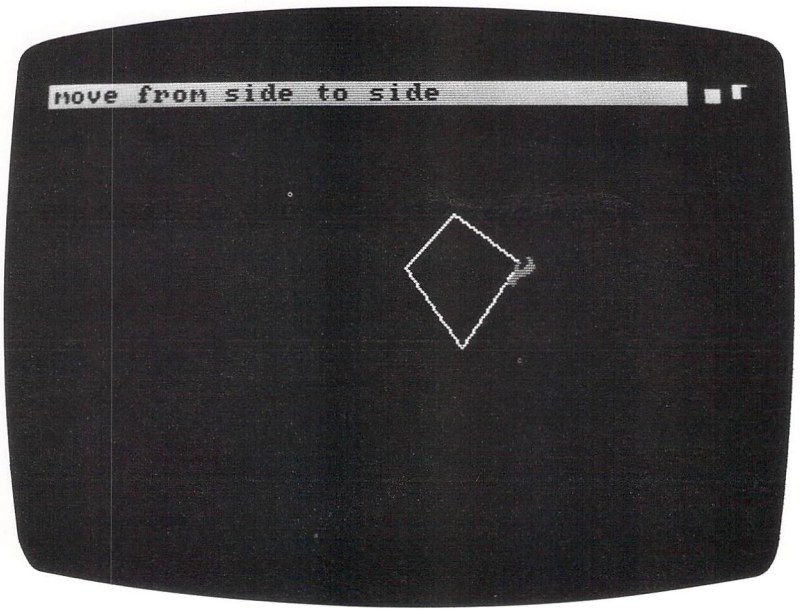


Figure 3.2 — Moving a Polygon Sideways.

- Place the cursor on the kite and hold down the Selection button. The shape will change to an outline. Now drag the cursor up and to the left. The kite will follow. Release the button.
- Select **replay tween** from the **Time** menu.

  The kite will move slowly from its starting point to the point where you dropped it after moving. As you can see, you did not have to worry about the image in between the beginning and ending points. Animator does that work.
- Select **next tween** from the **Time** menu.

We are now moving into the next segment of time in the animation.

- Select **rotate** and then **in plane** from the **Move** menu. The cursor will change to a crosshairs. Place the crosshairs in the center of the kite and click the Selection button.

  The cursor will change to a revolving circle and the crosshair will appear in the center of the kite. The center of the crosshair is the center around which the kite will rotate.

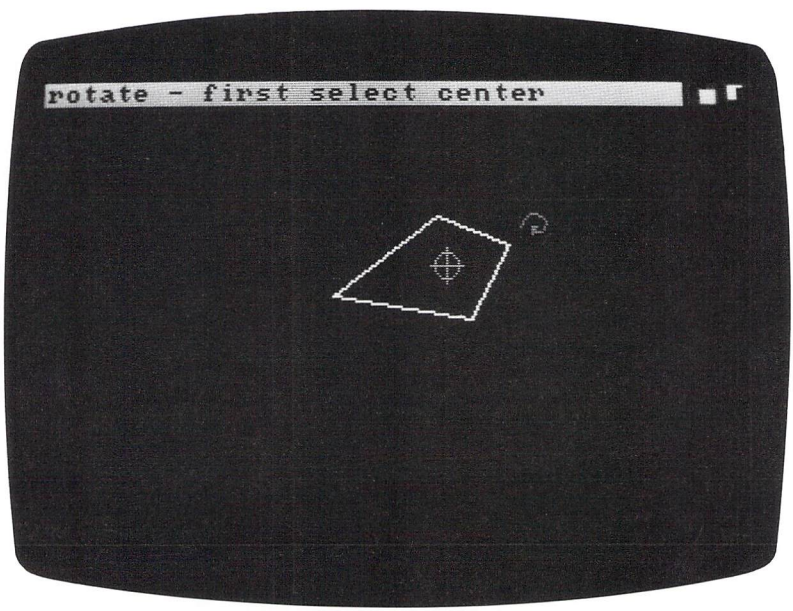


Figure 3.3 — Rotating a Polygon in the Same Place.

- Hold down the Selection button and drag the cursor in a circle. The kite will appear in outline and then move in a circle with the cursor. Spin the kite once and release the button.

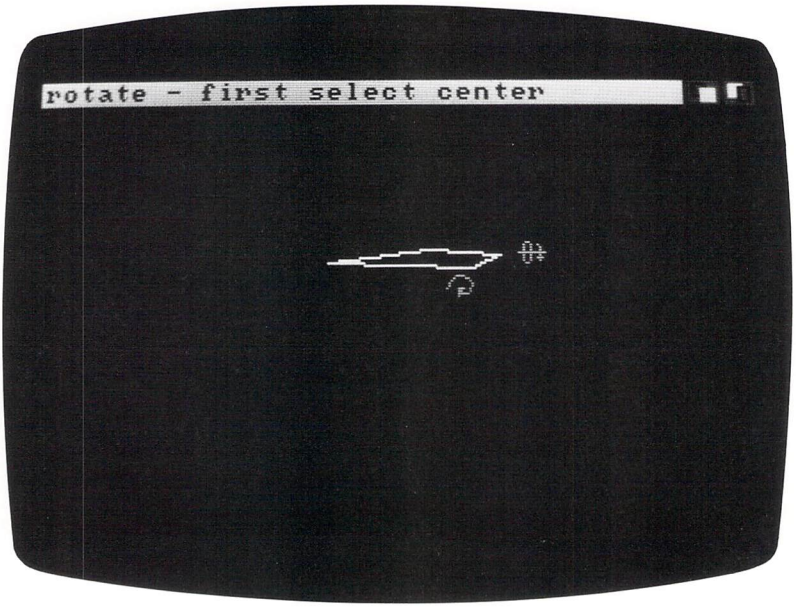


Figure 3.4 — Rotating a Polygon Around an X Axis.

- Select **replay all** from the **Time** menu.

  The kite will begin in its starting place and then move to the upper left part of the screen and spin once in a circle.

- Select **next tween** from the **Time** Menu.

- Select **rotate** and then **around x** from the **Move** menu. The cursor will appear as a straight horizontal line with a curving arrow through it. Place this at the bottom of the kite and click the Selection button once.

  The cursor will change to the rotation arrow and the **around x** cursor will appear on the screen. This is the axis around which the shape will rotate.

- Hold the left button down and drag the cursor toward the bottom of the screen. The kite will appear to flip on its axis. When the kite is pointing downward release the button.

- Select **replay all** from the **Time** menu.

Everything you've done should play on the screen. To see the loop repeat select **play loop** from the **Time** menu. To stop the loop click the Menu mouse button.

- Select **next tween** from the **Time** menu.
- Select **rotate** and then **around y** from the **Move** menu. The cursor will change to a vertical line with a curving arrow through it. Place this cursor to the right of the shape and click the Selection button.

  The cursor will change just as it did with the previous selection.
- Hold down the left button and drag the cursor to the right. The kite will flip to the other side of the center cursor. You may have to move the mouse up and down a bit to get the kite to regain its original shape (Later we'll go into more detail on how to adjust it back to the original).

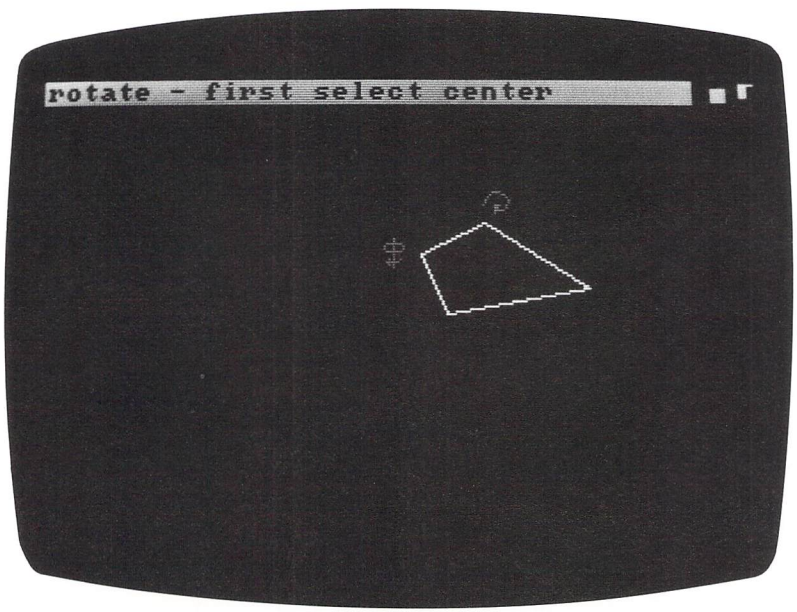


Figure 3.5 — Rotating a Polygon Around a Y Axis.

- Select **replay all** from the **Time** menu.

  It isn't necessary to select the next tween every time you move an object. We're only doing it in this example to clearly demonstrate what happens with each function.

- Select **next tween** from the **Time** menu.

- Select **size** from the **Move** menu. The cursor will change to the crosshair. Place the crosshair on the shape and click the Selection button once. The cursor will change to four arrows and the crosshair will appear in the shape to change. To make the shape larger, place the four arrows on top of the crosshair. Hold down the Selection button and drag the cursor away from the crosshair. When the shape is the size you want, release the button. To make a shape smaller, place the cursor away from the crosshair before holding down the button. Then hold the button and drag the cursor toward the crosshair. The kite will shrink in size.

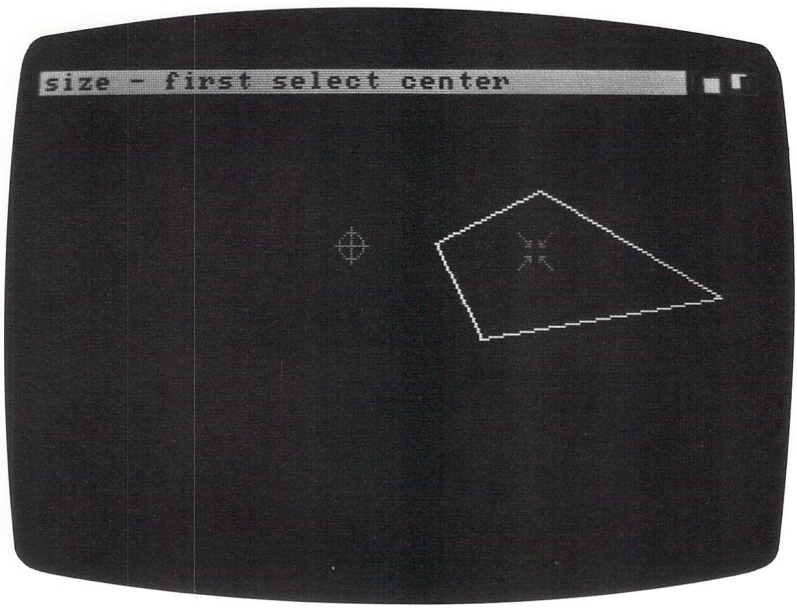


Figure 3.6 — Changing the Size of a Polygon.

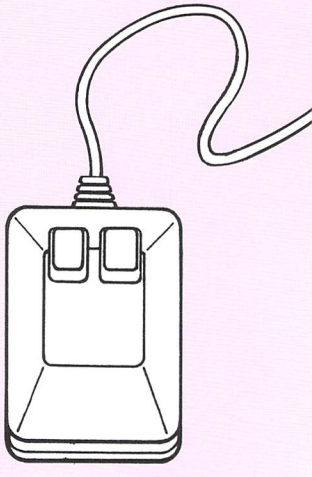
- Select **replay all** from the **Time** menu.

There are many more kinds of movement and any of them can be combined within a tween. These though, are the basics of how an animation can be created with Animator.

The next chapter will go into more detail about making shapes, and chapter 5 is a detailed account of how each type of movement works and how to combine them.

CHAPTER FOUR

Making Shapes



Making shapes is the most fundamental portion of Animator. The objects you create are the building blocks to the final animation. Even though back drops and other items can be loaded into an animation from a paint package such as Aegis Images, only the objects created from the **Create** menu can take full advantage of all the functions under the **Move** and **Select** menus.

As you read through this chapter try out the different shapes on the computer. Have fun with them and remember if you create something you don't want just select **undo** from the **Project** menu.

Lines

Lines are objects too. They use the same functions as any other shape created with Animator.

To create a line with the currently chosen color:

- Select **line** from the **Create** menu.
- Move the cursor to where the line should begin and click the Selection (left) button. If you decide you don't want a line, click the Menu (right) button.

- Move the cursor to where you want the other end of the line and click the Selection button again.
- If you want another line radiating from this point simply move in that direction. The cursor is still drawing lines. If you're done click the Menu button.

Polygons

There are two kinds of polygons available, filled and outline. Filled simply means a solid shape and outline is a series of lines that connect all the points of the polygon. A box made with four lines has different properties than a polygon made of four points. The box made of lines consists of four different objects (each line) but the polygon is one object.

To make a polygon with the currently selected color:

- Select **polygon** and then either **filled** or **outline** from the **Create** menu.
- Move the cursor to where you want the first point to be, click the Selection button.
- Move the cursor to the next point and click again. Continue in this fashion until you have the shape you want.
- When the shape is complete click the Menu button.

Circles

Circles are really polygons made of multiple points. You can add more points to a circle to smooth it out if you desire. Like polygons, there are two kinds of circles, filled and outline.

To make a circle in the currently chosen color:

- Select **circle** and then either **filled** or **outline** from the **Create** menu.

- Place the cursor where you want the center of the circle to be.
- Hold down the Selection button and drag the cursor away from its center. A circle will grow under the cursor.
- When the circle is the size you want release the button.

Stars

Stars come in two types, filled and outline.

To create a star in the currently selected color:

- Select **star** and then either **filled** or **outline** from the **Create** menu.
- Place the cursor where you want the center of the star.
- Hold down the Selection button and drag the cursor away from the center. A star will grow under the cursor. You can adjust the way the points are turned by dragging the cursor in a circle. The star will rotate on its center as you do so.
- When the star is the size and in the position you want release the button.

Blocks

Blocks are solid images known as rastors. They are not like the other objects in this menu. They don't use the more advanced **Move** commands though they can be pulled about the screen. They have one important advantage: they draw faster on a screen than regular polygons. Speed is important in keeping an animation smooth. Use blocks for patches of color you may need or square objects that do little movement.

To create a block in the current color:

- Select **block** from the **Create** menu.

- Place the cursor where you want the one corner of the block to be.
- Hold down the Selection button and drag the cursor to the opposite corner. A block will grow beneath the cursor.
- When the square is the proper size release the button.

Clone and Destroy

There are two other functions that affect objects as you create them. The first, **clone**, allows you to make exact duplicates of objects you have already created and the second, **destroy**, removes them from the animations present and future.

Clone

To clone an already created object:

- Select **clone** from the **Create** menu.
- Place the cursor on the object you want to clone. Click the Selection button once. To create multiple clones of an item, click the Selection button once for each item you want.

There are now two objects, one directly on top of the other. To see this better create a circle and then clone it. Now select **move** and then **sideways** from the **Move** menu. Place the cursor on top of the object you've cloned, hold the Selection button down and drag the cursor away. You will see that one object (the new one) was on top and follows the cursor while the other remains behind.

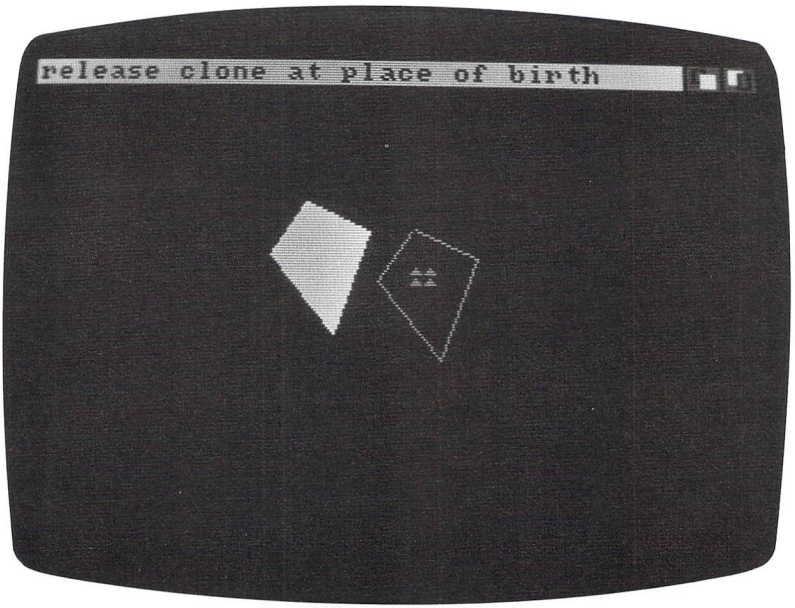


Figure 4.1 — Cloning a Polygon.

Destroy

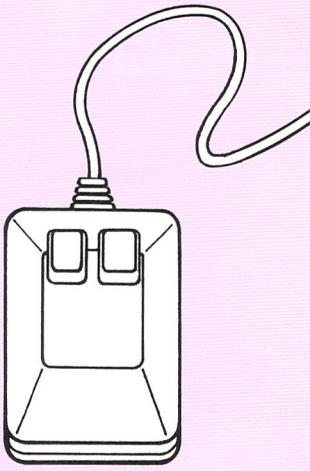
To remove an object from the current tween and any succeeding tweens you would choose **destroy**. It does not affect the object in any way previous to the current tween.

To destroy an object:

- Select **destroy** from the **Create** menu.
- Place the cursor on the object you wish to remove. The object will become highlighted. To cancel the **destroy** command click the Menu (right) button. To accept the command click the Selection button again.

CHAPTER FIVE

Movement



The movement functions are what make Aegis Animator so special. They provide a variety of ways to move objects about the screen and change their shape, color, or size.

Movement in Animator is metamorphic. In other words, the shape you are changing flows from its original form or position to the shape or position you change it to during the span of the current tween.

Some functions like **rotate—in plane** will record how many times you spin the object around a center, but most functions are only concerned with the beginning shape and the ending one. They then shift from one shape to the other when the tween is played.

As you read through this chapter try the different functions and experiment to be sure you understand how they work.

Sideways

The **sideways** function is one of the easiest to grasp and use. When in use the cursor becomes a hand and you may then drag objects anywhere about the screen with it. It is even possible to drag them off the screen.

To move an object sideways during the current tween:

- Select **move** and then **sideways** from the **Move** menu.
- Place the cursor on the object you wish to move.
- Hold the Selection button down and drag the cursor in the direction you wish the object to go. The object will be highlighted when it is selected and will follow the cursor.
- When the object is as far as you wish it to move during the current tween release the Selection button.

The object will move fast or slow depending on the distance it has to travel during the tween. You may move the object several times if you like during the current tween. The function is only concerned with the beginning position and the ending one. The object will move between these two points when the tween is played no matter where else it has been moved during the tween creation.

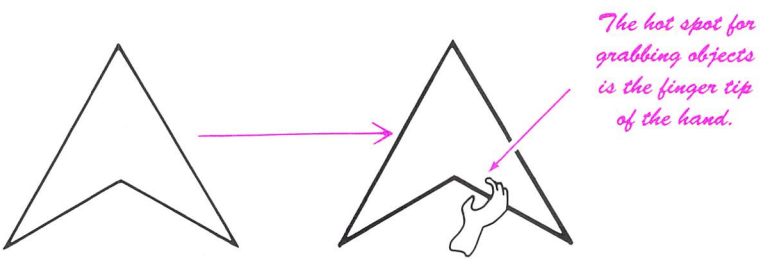


Figure 5.1 — Moving an Object Within the Same Plane.

In and Out

To understand the concept of **in** and **out** imagine a series of planes. Objects reside on these planes based on the time (which tween) they are introduced into the animation. So, if object B were dragged across object A (created first), object B would be on top. The **in** and **out** function allows you to change this order.

You use **in** to move objects behind others. Think of it as moving objects deeper into the screen, away from you. And **out** is used to move objects over others. Think of it as moving objects out of the screen, toward you.

To move an object in or out during the current tween:

- Select **move** and then **in** or **out** from the **Move** menu.
- Place the cursor on the object you wish to move.
- Hold down the Selection button and drag the cursor downward. The object will appear to get larger (**out**) or smaller (**in**) in response to the cursor movement.
- When the object is where you want it release the button.

The further you drag the cursor the further forward or backward the object will be moved. In other words, if you moved an object very far **in** you would have to drag the cursor even further to put another object behind it. Remember, it works like a series of planes—the further you drag the cursor the more planes you travel.

The best way to understand this function is to try it. Create two objects and then drag one so it partially covers the other. Now, play with the **in** and **out** functions moving the two objects in front of and behind each other.

Rotation

There are three types of rotation; **in plane**, **around x**, and **around y**. All three types use a center around which the rotation occurs. This center is created first and then the object is pulled around it. A variety of different things can happen depending on which type of rotation is used and where the center is located.

To get an object to turn like a wheel for example, you would place the center in the middle of the object itself. But to make the object move around in a circle you would place the center to one side and the object would spin around it.

In Plane

This function is rather like the **move—sideways** option. The movement only occurs in the plane the object currently occupies. The manner in which the object moves depends on where the center is placed. As described above, if it is placed in the middle of the object, the object will spin like a wheel.

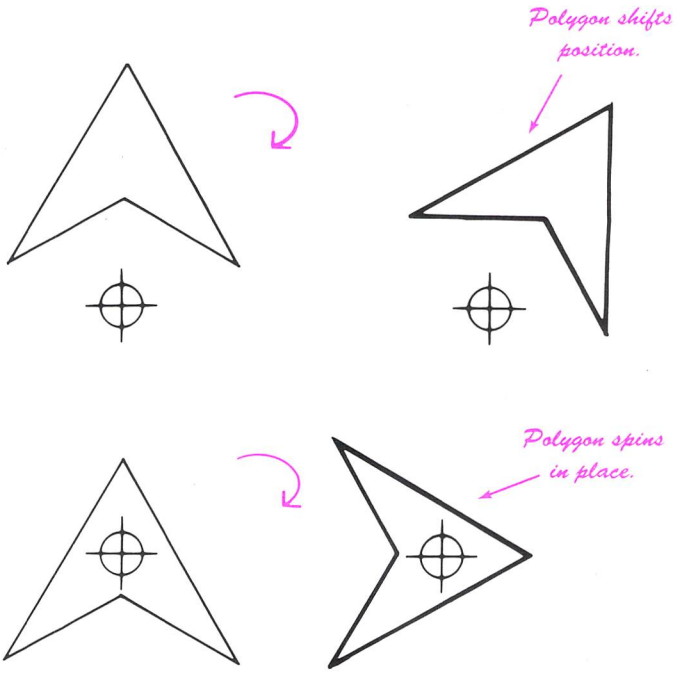


Figure 5.2 — Rotating a Polygon in Plane. This diagram shows the difference in movement depending on the placement of the center.

To rotate an object on the same plane in the current tween:

- Select **rotate** and then **in plane** from the **Move** menu.
- Place the cursor where you want the center of rotation to take place.

- Click the Selection button once. To cancel the command, click the Menu button.
- Hold the Selection button down and drag the cursor in a circle. The object will become highlighted and rotate around your chosen center.
- When you have rotated the object as much as you want in this tween, release the button.

The object's speed of rotation will depend on how far it must travel during the current tween. A greater distance will produce faster and more broken animation.

This function will remember the number of times you rotate an object around a center during a tween. Unlike most of the movement options, **in plane** does not move from a beginning point to an ending point only, but, remembers the entire movement and replays it.

Around X

The **around x** function will move an object around a point (the chosen center) horizontally. Depending on where the center is located, this can cause the object to change planes. If the center is located to either side of the object, **around x** will flip the object on its horizontal axis.

If the center is above or below the object, it will cause the object to spin around that point. The further away from the object the center is located the further it will flip the object inward or outward relative to its current plane.

To rotate an object on it's x axis in the current tween:

- Select **rotate** and then **around x** from the **Move** menu.
- Place the cursor where you want the center to be.

- Click the Selection button. If you want to cancel the command click the Menu button.
- Move the cursor onto the object you want to rotate. Hold down the Selection button and drag the cursor to make the object rotate.
- When the object is where you want it to be at the end of this tween release the button.

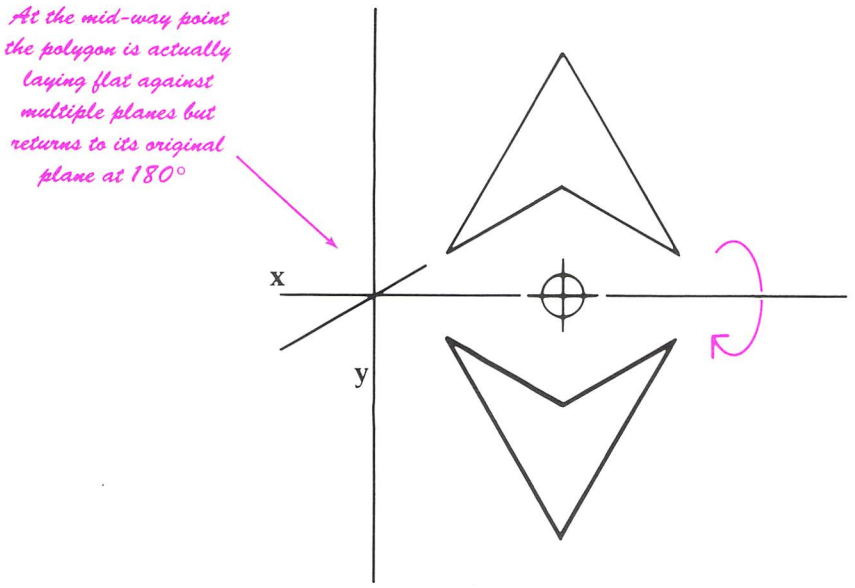


Figure 5.3 — Rotating a Polygon Around an X Axis.

Around Y

This function is the same as **around x** except that objects are moved on their vertical axis. The same rules apply as with **around x**.

To rotate an object on its vertical axis in the current tween:

- Select **rotate** and then **around y** from the **Move** menu.
- Place the cursor where you want the center to be.
- Click the Selection button. If you want to cancel the command click the Menu button.
- Move the cursor onto the object you want to rotate. Hold down the Selection button and drag the cursor to make the object rotate.
- When the object is where you want it to be at the end of this tween release the button.

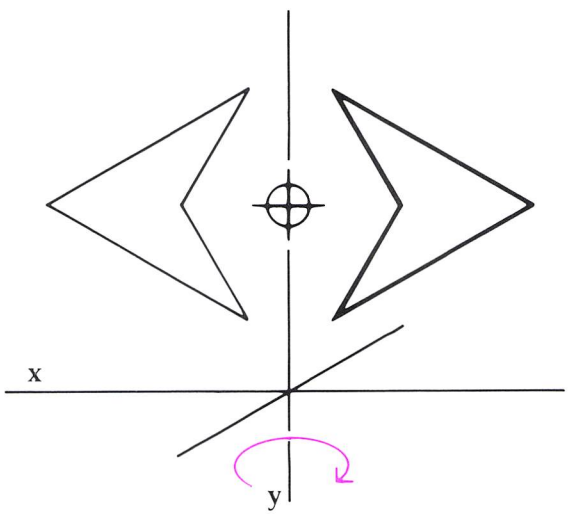


Figure 5.4 — Rotating a Polygon Around a Y Axis.

Size

The **size** option is concerned with the size of an object at the beginning of a tween and the new size at the end. Animator moves gradually from one to the other during the length of a tween when it is replayed. If **size** is used repeatedly on an object during the same tween it will try to balance the different commands.

Similar to the rotate functions, **size** has a center that is first located on the screen . The manner in which an object changes size depends on where you place the center. Placing the center in the middle of the object has the effect of keeping the object in place during its shift in size. Placing the center to one side of the object will cause the object to move as it changes. The direction the object moves depends on whether you are shrinking or enlarging and on which side of the object the center is placed.

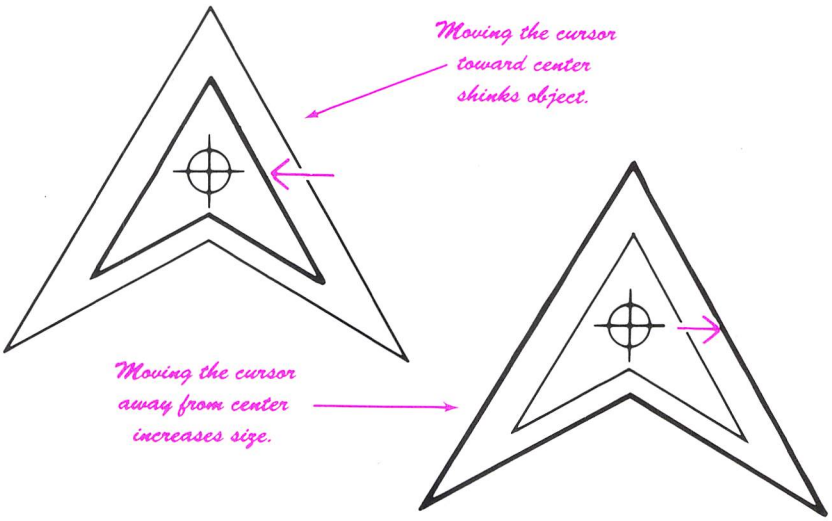


Figure 5.5 — Changing the Size of a Polygon. The manner of change depends on the direction the cursor is moved.

When you are shrinking an object it will always move in toward the center, and when you are enlarging it will move away.

To shrink an object in the current tween:

- Select **size** from the **Move** menu.
- Place the cursor where you want the center to be.
- Now move the cursor (no buttons) a little to one side of center.
- Hold down the Selection button and drag the cursor toward the center. The object will become highlighted and shrink in size.
- When the object is the size you want release the button.

To enlarge an object in the current tween:

- Select **size** from the **Move** menu.
- Place the cursor where you want the center to be.
- Keep the cursor on the center and then hold down the Selection button. The object will become highlighted.
- Drag the cursor away from the center and the object will grow in size. Move the cursor back toward the center to shrink the object.
- When the object is the correct size release the button.

The best way to understand how placing the center affects the result is to try it. Make a shape and try changing the size in different ways by placing the center in various locations.

Path

Objects can be given a specific path to follow during a tween. You do this by drawing the path on the screen made of points

and lines. It's not necessary to start the path at the object's current position. The object moves along the path from point to point relative to its starting position.

If you issue several path commands to an object during the same tween **path** will try to balance them. For example, suppose you have an object at the top of the screen in the middle. If you issue two different commands say, one down screen to the right and the other an equal distance down screen to the left, when you replay the tween the object will move straight down to rest in the middle of the screen on the bottom.

The **path** function is useful because it lets you see exactly what's going on. It also allows you to do things like moving objects off the screen, and if you wish, bring them back.

To move on object along a path in the current tween:

- Select **path** from the **Move** menu.
- Move the cursor onto the object you wish to move and click the Selection button once. The object will be highlighted to indicate it is selected.
- Now place the cursor in an area that provides the relative area you want and click the Selection button. A point will appear as if you were creating lines.
- Move the cursor in the direction of the first leg of the path. When the line is correct, click the Selection button again. When the tween is replayed, the object will move along a line like the one you just created relative to its starting position.
- Now move the cursor along the next leg of the path and select the next point. When the path is complete, click the Menu button.

Change Color

You can change the color of an object during any tween in an animation. The currently selected color under the **Color** menu

is active in the paintbrush when you select **change color** from the **Move** menu. To change it, simply select a new color from the menu.

When you change the color of an object, the change occurs at the beginning of the tween. For example, suppose you move an object about the screen and then change its color during a single tween. When the tween is replayed the object will change color first, at the beginning of the tween, and then it will execute the move commands you gave it.

To change the color of an object in the current tween:

- Select the color you want from the **Color** menu.
- Select **change color** from the **Move** menu.
- Place the tip of the paintbrush cursor on the object you wish to change and click the Selection button.

Change Type

There are three basic types of objects in Animator: filled, outline, and line. When you create an object by using one of the commands under the **Create** menu, you choose one of these types. The **change type** option gives you the ability to change the originally chosen type.

Like **change color**, **change type** occurs at the beginning of the tween in which it is used.

To change the type of an object in the current tween:

- Select **change type** and then either **to filled**, **to outline**, or **to line** from the **Move** command.
- Place the cursor on the object you wish to change and click the Selection button.

Morph

Morph is great! Morph is one of the features that makes Animator really unique and powerful. It allows you to change the shape of an object by moving the different points that make up the shape. You can even add new points to the object to adjust its shape further.

There are two separate tools in **morph**. The first is **loop**. Think of this as a lasso you can grab points with and drag them to where you want them. The other tool is **hook**. This is similar to **loop** but it can also add or delete points. For example, if you want to stretch one side of an object in a particular direction, **hook** will pull the line that makes up that side and place a point in the line allowing it to bend.

The **morph** function is concerned with how an object is shaped at the beginning of the tween and what it has been changed to. Any manipulations that occur in between are lost. When the tween is replayed, the object will shift gradually from its original form to the end result. For example, suppose you started with a triangle, and using morph changed it into a square. Then, after looking at it, you decided you really wanted a pentagon, and so changed it again. When the tween replayed, the triangle would shift slowly into the pentagon.

To change the shape of an object in the current tween:

- Select **morph** and then either **loop** or **hook** from the **Move** menu.
- Place the cursor on the object you wish to change and click the Selection button. The object will become highlighted and its connecting points will be visible.

If you chose loop:

- Place the cursor on the first point you want to change and hold down the Selection button. Now drag the cursor in the direction you want to go and the point will follow.

- When the point is where you want it release the button. Repeat this as many times as you need. When the shape change is complete, click the Menu button.

If you chose hook:

- You can place the cursor on a point or a line. Place the cursor where you want to change the shape and hold down the Selection button. Now drag the cursor in the direction you want to go.
- When the shape change is complete release the button. Repeat this as many times as you need. When you are finished, click the Menu button.

The hook can also delete points from an object. To do this:

- Place the hook cursor on the point you wish to delete.
- Drag the point to the next nearest point. The two points will become one point.
- Release the button.

Any point newly created with the hook will not delete points it encounters until the Selection button action (hold down) that created it is completed (releasing the button). This makes it easier to use the hook when working on small polygons.

Combinations

You may combine as many movements together on a single object as you like. The more **Move** commands you issue within a single tween however, the more complicated and possibly erratic the movement of the object. In most cases, Animator will try to balance all of the commands within a single tween. Sometimes the result may not be what you expected.

The way in which you combine movements has a lot to do with the outcome. Choosing the proper order in issuing

commands can play a big part. For example, there is an object we wish to move left to right on the screen and at the same time have it spin on its axis. If you choose **move—sideways** first, and then **rotate—in plane**, the object will swing low, close to the bottom of the screen, as it moves. This is because Animator is first moving the object to the right and then rotating it a fraction, and then moving it to the right, and rotating it a bit more. This type of combination forces it into a semi-circle path to reach the end position.

If you were to choose **rotate—in plane** first, and then **move—sideways** the result would be quite different. The object would travel on a nearly straight line left to right and spin in place as it traveled. This is because the process is reversed. Animator spins the object a fraction and then moves it to the right, spins a little more, and moves it to the right. The rotation process has no chance to move the object off the straight line course when Animator calculates its new position.

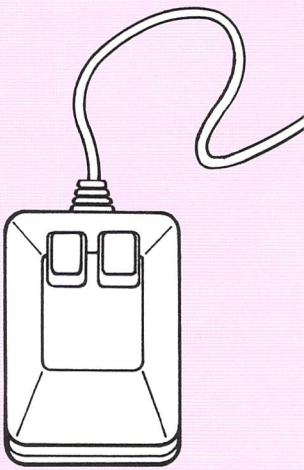
All of this may be a much more complicated explanation than you are interested in. The only real point behind it is—experiment. If a combination of movements doesn't produce the result you want, try reversing it or using a different set of tools. After a while familiarity with the subtle differences in each tool will produce the right combinations.

Sometimes **Move** combinations have to be issued to maintain the proper perspective. Animator can be a little approximate in maintaining a shape.

Rotating a shape for example can sometimes distort the image. To bring the image back into the proper perspective you can use the **morph** functions. The end result when the tween is replayed is the rotate and spin functions combining evenly to produce a perfect image.

Chapter 11, Animation Technique has more information on combining commands to produce polished results.

Select



So far we've only discussed moving or changing an object by itself. The **Select** menu gives you the ability to perform all of the actions described earlier on more than one object at a time or on only a portion of a single object.

If you pull down the **Select** menu you will see listed; **point**, **points**, **segment**, **polygon**, **polygons**, and **all**. These are the different levels of selection you may use. All of the actions described in Chapter 5 were using the **polygon** selection level. In other words, any **Move** functions selected occurred to a single object on the screen.

If you want to rotate several objects together, you need only choose **polygons** from the **Select** menu before you choose **rotate** from the **Move** menu. This would allow you to rotate the different objects around a common center with one command.

The point and segment functions let you perform actions against portions of a single object. You could grab two spikes of a star for example, and rotate them while the rest of the star remained untouched.

Point

This function restricts any **Move** commands to a single point of an object.

To change a point in the current tween:

- Select **point** from the **Select** menu.
- Select the **Move** command you wish to issue from the **Move** menu. If the command requires a center to be placed first (as in **rotate**), locate the center where you want it to be.
- Place the cursor on the point you want to change and hold the Selection button down. The object will become highlighted. The **Move** command will now operate in the normal manner.
- When the point is where you want it release the button.

Points

This is similar to **point** except that multiple points can be chosen. The points do not have to be on the same object. You may mark as many points as you want, and the action will be performed on all of them.

To change multiple points in the current tween:

- Select **points** from the **Select** menu.
- Select the **Move** command from the **Move** menu. If the command requires a center to be placed first (as in **rotate**), locate the center where you want it to be.
- Now place the cursor on the first point you want to change and click the Selection button once. The point will be highlighted to indicate it is selected.
- Place the cursor on the next point to change and select it. Continue this process until you have selected all the points you wish to change.

- Click the Menu button to finish. The **Move** command now functions in the normal manner.

Segment

The **segment** function operates on a portion of an object. The command works similar to the **points** function except a segment of the selected polygon becomes moveable.

To move a segment of a polygon in the current tween:

- Choose **segment** from the **Select** menu.
- Select the desired **Move** command from the **Move** menu.
- Place the cursor on the object to change and press the Selection button.
- Now place the cursor on the beginning of the segment you wish to change and click the Selection button again. A marker will appear.
- Place the cursor in the next position relative to the first marker to indicate the segment you want. Hold down the Selection button and drag the polygon segment in the direction you wish to go.
- When the shape change is complete release the button.

Polygon

If you have been using Animator as you read this manual then you are already familiar with this option. It allows you to perform a **Move** command on a single object such as a polygon. This is the default setting for **Select** when the Animator program begins.

To change a single object in the current tween:

- Select **polygon** from the **Select** menu.
- Choose the desired **Move** command. The command will operate directly on the object.

Polygons

Moving multiple polygons with a single command is similar to moving multiple points. Commands with a center such as **rotate** will move all the objects selected around a common point relative to their current position. The command is not limited to items you created with the polygon create function. Any object (line, star, circle, etc.) is considered a polygon and may be manipulated with this command.

To move multiple objects in the current tween:

- Choose **polygons** from the **Select** menu.
- Select the **Move** command from the **Move** menu. If the command requires a center to be placed first (as in **rotate**), locate that center where you want it to be.
- Now place the cursor on the first object you want to change and click the Selection button once. The object will be highlighted to indicate it is selected.
- Place the cursor on the next object to change and select it. Continue this process until you have selected all the objects you wish to change.
- Click the Menu button to finish. The **Move** command will now function in the normal manner.

All

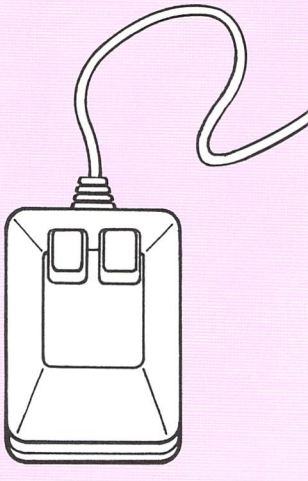
This version of the command chooses everything existing in the script including objects located off the screen. It does not affect paint pictures being used as backdrops.

To change all objects in the current tween:

- Choose **all** from the **Select** menu.
- Select the desired **Move** command. The command will function in the same way as with a single object, however, all the objects in the current tween will be affected.

CHAPTER SEVEN

Time and Color



Tweens and Time

Time in animations created with the Animator is controlled through a process known as **tweening**. Tweens can be thought of as segments of time in an animation. This is a rough analogy however, because the speed of a tween (the length of time it occupies an animation) can be adjusted. It can take less or more time to play out on the screen.

As an example of how tweens are used, consider an animation in which a ball bounces across the screen. To create the script you first create the ball, say in the upper left hand corner of the screen. Next, the ball is pulled downward at a slight right angle to the bottom of the screen (the floor the ball is bouncing on). This is the end of the first tween, and the command **next tween** would be chosen from the **Time** menu. Now the ball is pulled upward again at a right angle. This is the end of the next tween.

The natural bouncing of a ball has a slight pause in it at the top of a bounce where the upward momentum and the pull of gravity have a half second battle. To recreate this pause you need an extra piece of time at the top of the bounce. If you selected **next tween** and pulled the ball into its downward movement right away the effect would be to rigid. By inserting

an extra tween at the top of the bounce you have more time to play with.

In our ball example, the third tween has the ball at the top of its arc. To add just a bit more realism you might drag the ball just slightly to the right in the same way a real ball moves at its apex. Now the forth tween is begun and the ball is pulled downward again to the bottom of the screen.

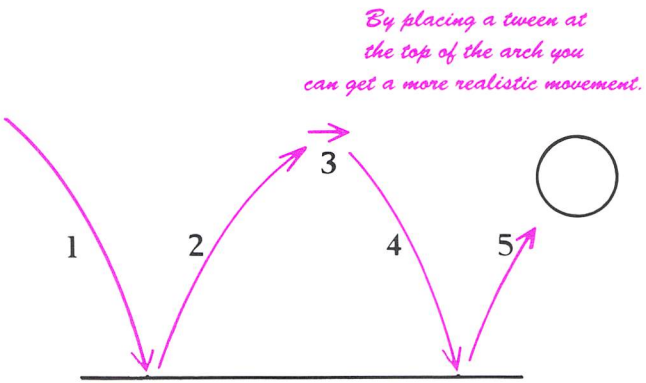


Figure 7.1 — An Animation of a Bouncing Ball. This can be used to describe tweens and the local time control.

If this sequence is replayed, the ball will seem to bounce but it will hover for a long time at the top of the arc. This is because all the tweens are exactly the same in length of time. To achieve the proper effect, the third tween (ball at top) needs to play faster.

Under the **Project** menu is the **time** function. Selecting this option displays a requestor window with several gadgets on it. These are the tools that allow you to control the speed of the animation. First, a quick tour and then we will return to the bouncing ball.

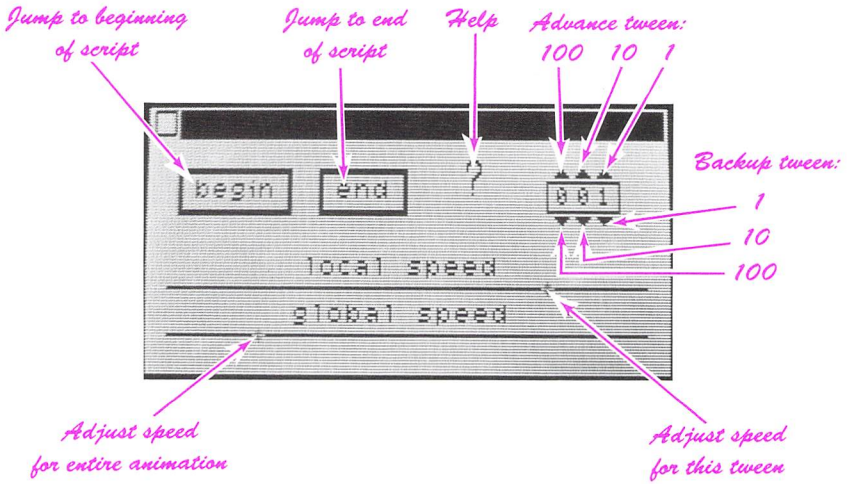


Figure 7.2 — The Time Requestor.

In the top left corner are two buttons labeled **begin** and **end**. Clicking the Selection button on these will display the current animation at its beginning or end respectively.

Just to the right of the button is a question mark. By clicking the Selection button while the cursor is on top of it you can get help dialog on any item in the menu. The cursor will change to a question mark. Simply place the cursor on any of the gadgets in the box and press the Selection button again. The menu bar will display a help message about that gadget.

In the upper right hand corner is a box displaying numbers. This is the current tween number. This gadget also allows you to jump to any tween in the animation. There are six buttons, three on top and three below the box. The three buttons on top advance you through the animation and the three below back up. Starting from the right moving left, the buttons jump one tween, jump ten tweens, and jump one hundred tweens.

At the bottom of the window are two sliders, one labeled **local speed** and the other **global speed**. Local speed is the speed of the current tween (displayed in the box above), and

global speed is the speed of the whole animation. Which brings us back to the bouncing ball.

In order to smooth out the animation the third tween needed to play out faster. By using the **jump to tween** buttons the third tween can be displayed on the screen. When the box reads **003** the third tween is the current tween and the **local speed** slider is controlling its speed. By grabbing the slider with the cursor (holding the Selection button down) and moving it to the right the speed of tween three is increased.

Now if the animation is replayed the ball more closely approximates the bouncing of a real ball. The animation can be perfected even more by increasing the global speed a little and then adjusting the speed of each tween to get just the right effect.

Color Control

There are many different effects that can be accomplished with the use of color changes in an animation. A sunrise can be simulated by slowly fading all the colors in the palette from dark to light across several tweens or a camera fade out can be accomplished by the same process in reverse. A backdrop made in a paint program can have a stream that appears to be flowing by cycling the colors of the water, or a water wheel that turns by alternating the colors of the spokes.

Aegis Animator gives you the full range of colors available to the Amiga with any 32 colors available at the same time on the screen. The colors can be shifted gradually or changed instantly with the use of various tools.

There are three locations for color commands in Animator. The first is the color palette located under the **Color** menu. Also the **Move** menu has a **change color** command for changing the color of objects in an animation, and under **Project** there is a **color** command that displays a variety of color control tools.

By selecting **color** from the **Project** menu a requestor window will appear with a color bar across the bottom and several gadgets on it. The gadgets that dominate the left hand side of

the window control the actual shade of a color. By manipulating these tools you can produce any of 4096 different colors.

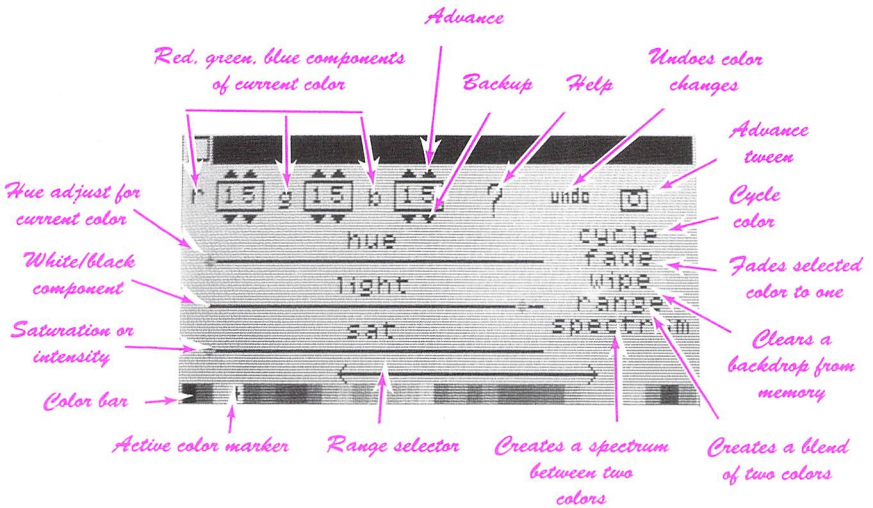


Figure 7.3 — The Color Requestor.

There are three boxes at the top displaying numbers. To the left of each one is a letter; **r**, **g**, and **b**. These stand for red, green, and blue. The numbers may range from 0 to 15 and describe the amount of each color to blend into the mix making up the currently selected color. The current color is the one with the crosshair in the color bar at the bottom. If the numbers in **r** and **g** were set to 0 and **b** were set to 15 the selected color would become a very intense blue. To get a yellow you would set **b** to 0 and **r** and **g** to 15.

These boxes have buttons above and below in the same manner as the tween control device in the time window. The buttons on top advance the numbers while the ones below subtract from the total. The button on the left jumps 10 while the one on the right jumps 1.

Below these gadgets are three sliders. They also control the actual shade of a color. In fact, if you play with the r-g-b controls you will notice the slider controls moving in response to the changes you make. This is because they are duplicate controls. They're just another way of doing a similar job. Sometimes it's easier to use one set of tools over another for a particular task. Instead of controlling the amount of red, blue, and green however, the sliders control the hue, light, and saturation.

The **hue** is the particular shade in the spectrum, light blue for instance. **Light** adjusts the black and white component of the color or the brightness. Moving the slider to the left will darken the color until it at last fades to black while pulling it to the right will brighten the color eventually turning it white.

Saturation can best be described as the vibrancy or intensity of the color. Moving the slider to the left will produce muted or softer tones while moving it to the right will create bright, almost day glow colors.

In the upper right corner are three functions not directly related to color control. The first is the help question mark. This icon works in the same manner as the one in the time window. Use the Selection button to choose help and then press the button again with the cursor located on the item you need help with. The menu bar will display a brief description of the function.

The next item to the right is **undo**. This function will undo the last operation you performed in the color window. It's there for the inevitable mistake.

The last item to the right is a camera icon. Pressing the Selection button while the cursor is on this will advance the current tween by one.

Below the camera icon are five color functions. They perform different actions such as cycling colors through an animation, creating a range of colors in the palette, or fading a range of colors to create a blackout.

All of the commands except **wipe** use a range selector located just above the color bar. The selector is the line with arrows on each end. The arrows indicate the beginning and ending colors

of a range, and the different commands perform actions against the colors located within the range. To adjust the range, the arrows may be moved by holding the Selection button down while the cursor is on an arrow and then dragging the arrow to the desired position.

If you drag the arrows past each other you will notice they flip in to point at each other instead of away. Any functions performed with the color range will now occur in reverse of its normal order.

A Note About the Color Bar

The color bar and the color menu are made up of 32 different colors. Imagine the boxes containing the colors as ink wells. Each ink well has a number. An object created with a certain ink well may be blue, but with a few adjustments of the color controls that well can be changed to red. The object created with that particular ink well would also change to red. This is because Animator sees objects as having a certain color number. You can change that number with the **change color** command from the **Move** menu or you can change the shade of the actual well itself with the commands in the **color** window.

The functions described below act in terms of color numbers. Color changes affect anything displayed whether it's a backdrop loaded in from a paint program or an object created within Animator.

Cycle

The **cycle** command allows you to cycle different colors in an animation. For example, you may want objects on the screen that are currently green to slowly change to red during a particular tween, or portions of a backdrop, normally static, to appear to move.

Cycle will evenly fade from the first color to the next chosen color across the life of a tween. If many colors are chosen during

one tween, Animator will try to display them all. If the time span is very short and the number of color changes great then the changes will occur very fast.

To cycle colors in the current tween:

- Select **color** from the **Project** menu.
- Adjust the range selector to reflect the series of colors to cycle. The colors will move one to the right each time you click on the cycle command. So if your color range is red, green, yellow, blue, and you click on the command all red objects will shift to blue, greens to red, yellows to green, and blues to yellow.
- Click the Selection button on **cycle**.
- If you wish more than one cycle during the current tween click the command again. If you wish to cycle a different set of colors during the current tween adjust the range selector and choose the command again.
- If you wish the cycling to continue across multiple tweens, choose next tween by clicking with the selection button on the camera icon in the upper right corner of the window, and then repeat the cycle process.

Fade

This option allows you to perform actions such as fade to black. Animator will gradually fade a range of colors to a single color across the life of the current tween.

To perform a fade in the current tween:

- Select **color** from the **Project** menu.
- Adjust the range selector to indicate the colors you wish to fade.

- Choose the color to fade to (black for instance in a fade out) by clicking the Selection button with the cursor on the desired color in the color bar. A cross mark will appear in the color to indicate it is selected.
- Choose **fade** with the Selection button.

Wipe

The **wipe** function clears a backdrop off the screen and from memory.

To clear a backdrop at the beginning of the current tween:

- Select **color** from the **Project** menu.
- Choose **wipe** with the Selection button.

Range

The **range** command is for adjusting the color bar to produce a series of colors that blend two tones together. The arrows of the Range Selector indicate the beginning and end of the series as well as the two tones to mix. Blending two colors like red and green together will produce brown tones in the middle. The longer the distance between the two colors the more gradual the shading will be from color to color.

To produce a range of colors in the color bar:

- Select **color** from the **Project** menu.
- Adjust the range selector to indicate the beginning and end of the color range.
- Choose **range** with the Selection button.

Spectrum

This option is similar to **range** but the end result is very different. Instead of blending, the color series produced is more like a rainbow effect between the two selected tones.

Reversing the Range Selector (arrows pointing inward) will produce different results with the same two colors.

To produce a spectrum between two colors in the color bar:

- Select **color** from the **Project** menu.
- Adjust the range selector to indicate the beginning and end of the color spectrum.
- Choose **spectrum** with the Selection button.

Saving and Loading a Palette

Once you have created a palette exactly to your liking you can save it as a color bundle and load it into an animation at any time. This way you don't have to recreate a particular palette every time you want to use it, or you can suddenly change a color scheme in the middle of an animation to create different effects.

The original set of colors that come with Animator are stored in a color file named **start**. If at any time you wish to reset your color palette to the original colors simply load this file.

To save a color palette as a file on disk:

- Select **storage** from the **Project** menu. The storage requestor window will appear.
- Choose **col** from the suffix (middle) column. The left column will change to display any color files already saved. The name in the lower right corner of the window is the current active file name. If you wish to use this name skip to the next step. Place the cursor on the name and press the Selection

button. Type in the name you wish to save this color file under and press RETURN.

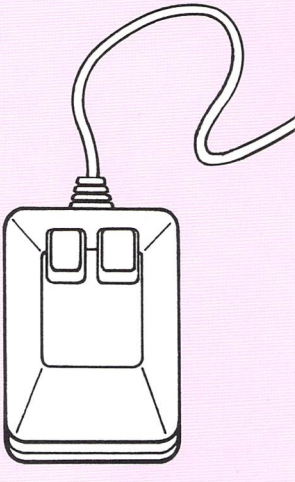
- Select **save** from the action (right) column. The current color palette will be saved to disk.

To load a previously saved color palette into an animation at the current tween:

- Select **storage** from the **Project** menu. The storage requestor window will appear.
- Choose **col** from the middle column. The left most column will display a list of all available color files in the currently logged directory.
- Place the cursor on the file you wish to load and press the Selection button.
- Select **load** from the action column. The color palette will load into the computer. Any time the animation reaches this tween the load action will be performed.

CHAPTER EIGHT

The Storyboard



The storyboard is more like a film editor than a traditional storyboard. It has nine slots displayed as windows that can each hold a separate animation. It is here you can cut out unwanted tweens or splice animations together.

It is a good idea to save your animation to disk before going to the storyboard since it does not have an undo feature. If you accidentally delete an animation it's gone for good. Also, memory can be used up very fast when moving animations around and it's possible you can run out. Be sure to check the **status** command regularly to keep an eye on how much memory is left.

You reach the storyboard by choosing **storyboard** from the **Project** menu. The screen will change to display nine windows. One window will be highlighted. This is the current animation you are working on.

Storyboard has two main menu selections, **Project** and **Edit**. Under **Project** are the miscellaneous tools used in this portion of the program. The first, **go into** is the default setting when storyboard is first entered. The cursor will be shaped as **into**. By placing the cursor on one of the windows and pressing the Selection button you will be returned to the creation portion of the program working on that particular animation. To get the **into** cursor, choose the **go into** command at any time.

Speed

The speed command is similar to the **time** function. Choosing this command displays a requestor window with two command options and a slider control. The slider controls the overall speed of the current animation. To increase the speed move the slider to the right.

Above the slider are two options, **do fills**, and **ghost mode**. The **do fills** option is the default setting. It means all objects created as filled objects from the **Create** menu will appear that way on the screen. If you were to choose **ghost mode** instead the objects would appear as wire frame graphics. This is sometimes useful for getting at objects that are temporarily behind other ones, or for getting a better view of what's going on. The **ghost mode** can also be activated from the **Time** menu of the creation portion of the program.

Status

The **status** command is the same as under the **Project** menu in the creation portion of Animator. It displays a message in the menu bar about the current amount of free memory left.

Exit

The **exit** command will exit Animator and return control to the Workbench.

Edit

Under the **Edit** menu are the real tools that make the storyboard so useful. The **splice**, **cut**, and **delete** commands allow you to cut out, and insert tweens in an animation. You can also splice animations together.

Splice

The **splice** function will pick up an animation in one window and attach it to the end of an animation in another window. If the window is blank then the animation is merely copied into the new window.

To splice an animation onto the end of another:

- Choose **splice** from the **Edit** menu. The cursor will change to resemble a film splicer.
- Place the cursor on the animation you want to pick up and press the Selection button once.
- Place the cursor in the window where you want the animation to be added. Press the Selection button again.

Cut

The cut feature is similar to splice in that it picks up an animation and puts it down in another window. Whereas **splice** copies though, **cut** actually cuts. It will chop an animation at the current tween and pick it and all future tweens up and lay them down in the new window.

The tween where the cut occurs can be chosen by selecting a specific tween from the **time** window before entering **storyboard**, or by using **activate** (see below).

This command can be used for cutting unwanted tweens off the beginning or end of an animation. When used together with the splice command it can be used to cut tweens out of the middle of an animation.

To cut an animation at the current tween:

- Choose **cut** from the **Edit** menu. The cursor will change to represent a pair of scissors.
- Place the cursor in the window where you want to pick up an animation and press the Selection button.

- Now place the cursor where you want to put the cut strip and press the Selection button again.

Delete

The **delete** option is used to clear a window. It is functionally the same as choosing **new script** from under the **Project** menu in the creation portion of the program.

To delete an animation from memory:

- Choose **delete** from the **Edit** menu. The cursor will change to the destroy icon.
- Place the cursor on the window to clear and press the Selection button.

Activate

This command is similar to **play loop** from the **Time** menu. It allows you to run and stop the animations in the different windows. All nine windows can run at the same time though there may be some slowing of the animations if that many are playing at once. The command can also be used to stop a script at a certain point for editing.

To play an animation:

- Choose **activate** from the **Edit** menu. The cursor will change to represent a movie projector.
- Place the cursor on the window to activate and press the Selection button.

To stop an animation place the cursor on the window and press the button again.

Storyboard and Paint Program Images

If there are images loaded from outside Animator on the screen when you go to the storyboard they will not be visible. Animator has no way to shrink these items to display them in the limited space of the slots so the items are not shown. Masks and windows can be viewed in **ghost mode**. The items will appear as boxes in their relative positions on the screen. The **ghost mode** option can be issued from the **speed** function in **storyboard** or from the **Time** menu in the creation portion of the program.

When you return to the creation portion of the program the items may still be missing. This is because the current tween is not the tween where the items are to be loaded into the animation. To have the items reappear on the screen simply replay the animation or jump to the specific tween where they are loaded (the beginning for example).

Memory Panic—System Going Down

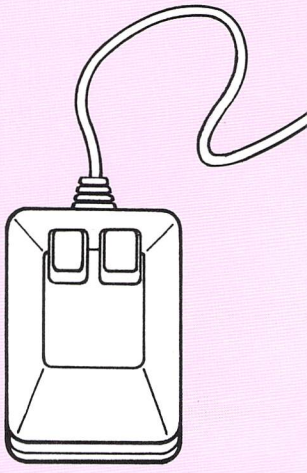
Animator is a big program. Animations are complex things that require a lot of memory. The situation is bound to occur at some point where you run out of space particularly in the storyboard where large chunks of memory get used up quickly. Fear not! Animator will catch itself if a fatal error in memory allocation takes place and give you the chance to save your work.

The message "MEMORY PANIC—SYSTEM GOING DOWN" will appear in the menu bar and a requestor will be displayed on the screen asking if you wish to save the storyboard. If you respond with a **no** the program will exit to the workbench and everything in the storyboard will be lost. Anything you saved previously on disk will still be there.

By responding with **yes** to the requestor Animator will begin saving a series of files on the disk. Each file will be titled **ani_panicX.script** where **X** is the number 1 through the number of animation windows you had operating in the storyboard. To begin again, reload Animator, and using the storage command load the scripts you wish to continue working with.

CHAPTER NINE

Using Images Created Outside Animator



Aegis Animator allows you to bring pictures created with paint programs such as Aegis Images into an animation. Any program which supports IFF (Interchange File Format) can be used. There are three types of images used: backdrops, windows, and masks.

Backdrops

A backdrop is an image that resides behind any animation created. It acts as a setting for the script to unfold before and with the **cycle** command it can perform some limited animation of its own. The image does not move. It appears when you issue the **load** function from the storage requestor, and resides in the back until the end of the animation or until you issue a **wipe** command from the color requestor. Backdrops can be loaded in at any time in an animation, and you can load one backdrop in while another is already present. As soon as the load is finished the first picture will be wiped and replaced with the new image.

Backdrops are filed on disks as **.pic** files, which means they have that extension on their file name (filename.pic). The storage requestor window has a **pic** function in the center column for handling backdrop files.

To load a backdrop into an animation at the current tween:

- Select **storage** from the **Project** menu. The storage window will appear.
- Choose **pic** from the suffix (center) column. A list of pictures available in the current directory will be displayed. To change directories select **cd** (change directory) from the action (third) column, and enter the new directory name at the prompt. Be sure to press RETURN after the directory name.
- Select the name of the picture you want to load from the first column and then choose **load** from the third column. The backdrop will be loaded and displayed.

Windows

To use the jargon, windows are bit map images or rasters that can be moved about on the screen. In non-computerese they are objects you capture from a painting with the use of a frame. The frame is saved to disk. The window can then be loaded at any time in an animation and moved about on the screen. This allows you to create objects with much more detail than is possible with the polygon creation tools of Animator.

Heres how you do it. First you load a painting (pic file) in which you have created the parts you want to use. Then you capture those items as windows (described below in detail) and save them to disk. Next, clear the painting from memory and move into the animation to the tween where you want the items to appear. And last, load the windows using the storage commands.

If you are using the Aegis Images paint program you can save windows directly from there with the **save—window** command.

Windows are square polygons that have limited movement. If the object you captured is on a background the same color as the animation background then the square edges will blend in and just the object will be seen. The only **Move** commands

available to a window are **move—sideways**, **move—in**, **move—out**, and **path**.

Raster objects like windows and masks (below) draw faster than filled polygons. The more objects you create as rasters, the smoother the animation will appear.

To save an object on the screen as a window:

- Choose **storage** from the **Project** menu. The storage requestor window will appear.
- Select **win** from the middle column. A list of available window files will appear on the left.
- In the lower right corner is the current active file name. If this is the name you wish to use go on to the next step. Click the Selection button with the cursor located on the file name. The name will disappear. Type in the name you wish to save the window under, and press RETURN.
- Choose **save** from the last column. The storage window will disappear and the cursor will become a crosshair.
- Place the cursor on one corner of the desired capture area. Hold the Selection button down and drag the cursor to the opposite corner. A highlighted frame will appear around the object to capture.
- When the frame is where you want it release the button. The file will be saved.

To load a window into an animation at the current tween:

- Choose **storage** from the **Project** menu. The storage window will appear.
- Select **win** from the middle column. A list of available window files will appear to the left.
- Choose the name of the file you wish to load.

- Select **load** from the right hand column. The file will be loaded and the storage window will disappear.
- The window will be on the screen highlighted with a frame. Place the cursor on the object and hold down the Selection button. Move the window to the position where you want the object to appear, and release the button. Any subsequent changes in position from now on will appear as movement in the animation.

Masks

Masks are similar to windows except that they are only one color. The currently active color is used at the time of their creation. Masks can be made from objects in paint files like windows but their best use is in duplicating polygons.

If there are polygons in your animation that will not need any of the fancy rotation, morph, or size **Move** commands then a good thing to do is change them into masks. Remember masks, like windows are rasters and draw faster than polygons. This speeds up and smooths the animation.

When a mask is captured, any background color captured inside the frame becomes transparent when the mask is used in an animation. This is different than with a window. A window's background blends in when possible, but a mask's background colors are not captured with the image. Thus, it is just the image itself that appears as a mask.

To save an object (polygon or paint image) as a mask:

- Choose **storage** from the **Project** menu. The storage requestor window will appear.
- Select **msk** from the middle column. A list of available mask files will appear on the left.
- In the lower right corner is the current active file name. If this is the name you wish to use go on to the next step. Click the Selection button with the cursor located on the

file name. The name will disappear. Type in the name you wish to save the mask under, and press RETURN.

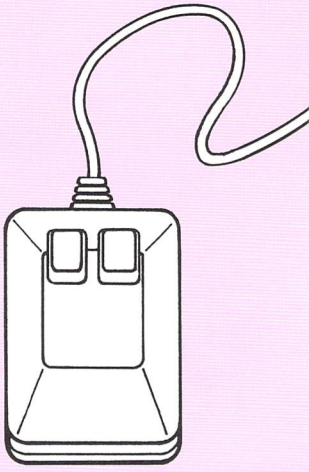
- Choose **save** from the last column. The storage window will disappear and the cursor will become a crosshair.
- Place the cursor on one corner of the desired capture area. Hold the Selection button down and drag the cursor to the opposite corner. A highlighted frame will appear around the object to capture.
- When the frame is where you want it release the button. The file will be saved.

To load a mask into an animation at the current tween:

- Choose **storage** from the **Project** menu. The storage window will appear.
- Select **msk** from the middle column. A list of available mask files will appear to the left.
- Choose the name of the file you wish to load.
- Select **load** from the right hand column. The file will be loaded and the storage window will disappear.
- The mask will be on the screen highlighted with a frame. Place the cursor on the object and hold down the Selection button. Move the mask to the position where you want the object to appear, and release the button. Any subsequent changes in position from now on will appear as movement in the animation.

CHAPTER TEN

Storage



If you've been using Animator as you read this manual or if you've looked at some of the demos then you have already worked with the **storage** command. There are six different types of files used by Animator and the window displayed by the **storage** command from the **Project** menu is how you manipulate them.

When you select the command, the storage requestor window is displayed and the menu bar holds the message **select title, suffix, action**. In the window itself are three columns of text. Each column aligns with the names in the menu bar. The first column holds the titles of all available files of that type (suffix) in the currently logged directory.

The second column is the file type or suffix. As mentioned above there are six of these; **script**, **pic**, **win**, **msk**, **poly**, and **col**.

A script file is the actual animation. It contains the commands that tell Animator how to assemble and display an animation.

Pic files are pictures saved from other programs such as Aegis Images to be used as backdrops in an animation. **Win** and **Msk** stand for window and mask. These file types are described in detail in the chapter on using images created outside Animator (Chapter 9).

Poly files describe individual polygons or objects. These can be saved or loaded at any time. This allows you to create polygons separately from an animation and then load them in at the proper time. The steps to save or load a polygon are the same as saving a mask. Using mask files is detailed in the chapter on using objects created outside Animator (Chapter 9).

Col files are color bundles. With this option you can save and load an entire palette setting at any time in an animation. This option is described in more detail in the chapter on time and color (Chapter 7).

These suffix are added to a file name when it is stored on a disk. So a demo animation would have the full name on disk of **demo.script** and a backdrop for that same animation may have the name **demo.pic**.

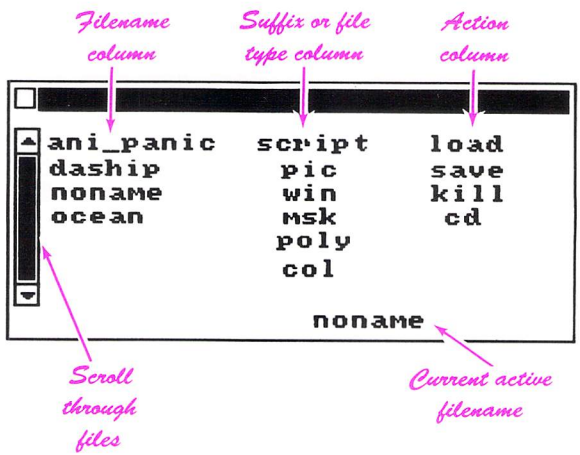


Figure 10.1 — The Storage Requestor.

The third column of the storage window is the action column. It performs the indicated action on the currently logged file. In the lower right hand portion of the window is the name of the current file. To make any file in the first column current just place the cursor on it and press the Selection button.

To enter a new name (for saving a new file for example) place the cursor on the current file name in the lower right

hand corner and press the Selection button. The name will disappear and you can then type in the new name. Be sure to press RETURN when you're through.

When you're ready to perform an action on a file select the action from the third column. The **load** command will bring the file off the disk and into the computer. The **save** command puts a copy of the file from the computer to the disk.

To delete a file from the disk (permanently) select **kill**. A requestor will appear first to be sure you really want to delete this file.

The **cd** in the action column is for **change directory**. This allows you to access other disk drives or sub-directories on the same disk.

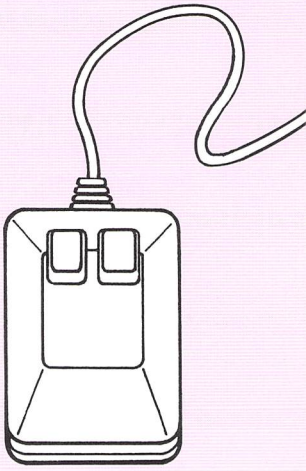
To change directories:

- Select **cd** from the action (right most) column of the storage window. If you want to cancel the command at this time press the Menu button.
- Type in the name of the directory you wish to access and press RETURN. Animator will read the files in the indicated directory and then display their names in the title (left most) column.

To close the storage window press the Menu button or place the cursor in the close box of the window and press the Selection button.

CHAPTER ELEVEN

Animation Technique



Okay, here's the part of the manual where we tell you all the good stuff. How to put everything together to make an animation, tips to remember to make the final result look clean, and shortcuts. First a few hints on getting the best results.

Be aware of the media you are working in. Animator uses low resolution graphics (320X200) and this puts limitations on the images you can create. When you make an object with one of the **Create** functions you will note it is made up of a series of horizontal colored lines on the screen. Separating each line is a thin line of the background color. This is one of the controlling features of the media you are working in. Objects look best when made of clean, balanced lines.

For example, choose **polygon—filled** from the **Create** menu, place a point on the screen and stretch a line from the point. If the line is perfectly horizontal it appears unbroken and clean. Dragging the line at an angle up or down breaks it into a series of steps. Depending on the location of the end point these steps can be either even or ragged. A perfectly clean line appears every 45 degrees around the point, but even points off the 45 degree mark can be moved about a little until each step is even with its neighbors. Even lines give an animation a polished look.

Also, have patience, and pay attention to detail. Like any art form, animations created with Animator will look better with the more time and effort you put into them. Don't try to produce *Gone With The Wind* the first time out. Play a little first to get the feel of the system. Practice with the different tools and with building certain illusions. That is what animation is all about, creating illusions.

Planning

Some animations can just be created on the fly without any advance work, but the more complicated ones will need a lot of preparation. Backdrops, text scrolling on the screen and paint images that move about all need work done in a paint program like Aegis Images first.

Another example of necessary planning is an object suddenly appearing from behind another. Polygons exist on successive planes **outward** from the screen as they are made. So if you suddenly want your space creature to grow antennae smoothly they need to have been made when the creature was originally created and kept as an inactive polygon behind the rest of the creature until needed. This needs a little forethought since going back later and inserting the antennae will always have them appearing on top of what is already there.

One way to dance around some of these problems is with the **save** and **storyboard** features. You can create objects independent of the end animation and save them to disk as a polygon, window, picture, or mask. Then go back and build the final animation piece by piece. When each element is needed in the animation, load it from the storage window.

The storyboard provides a similar capability without the flexibility of bringing in a single object at the right moment. Starting tweens can be perfected in one window and then the preliminary adjusting movements cut out as a tween and the final product moved into a new window ready to begin. Or, using the space creature antennae example, a copy of the current

animation can be spliced into a clean window and the antenna added in that version of the script. Adjusting movements can be made such as a **move—in** etc. When the script is perfect a **next tween** is issued and that last tween is cut out and pasted back onto the original version. Everything is the same as if you had just advanced the tween except there is now a set of antennae behind the creature.

Speed

Planning is also useful in getting smoother animations. Speed is the main consideration in avoiding jerky movements. It takes Animator longer to draw a polygon than a mask or a window and color changes take even less of the processing time.

For these reasons you want to animate as much as possible with masks and windows and save the polygons and other shapes for key roles and enhancements to the raster (mask and window) images. For example, a tree can be made by creating the trunk and some leaves in Images and the rest of the leaves or branches that need to move in Animator. The Animator leaves are drawn right on top of the rastor image and even though just a few of the leaves move it appears as if the whole tree is animated. Add to that a subtle color shifting of rastor leaves and the effect is complete with only a few polygons needed.

Polygons need to be redrawn by the computer during every screen refresh even if they didn't change position or shape so the fewer on the screen the better.

Another way to get good control of the speed and smoothness of the animation is to use plenty of tweens. The speed is adjustable in each one so you have more control over individual movements. This is particularly true of morphing type movements where the in-between shape might not be exactly what you want.

Upstream

Once an animation is complete or nearly complete you'll want to go back and watch the whole thing several times, looking for places in need of minor adjustments. When changes are made in the middle of an animation it's called upstream editing. This means changes are being made to an object which already has movements assigned to it sometime in the future.

Move commands assigned to objects are relative things. If you have moved an object a half inch to the right in tween 10 and then go back and change tween 5 by moving the same object to the right a half inch the overall movement by the end of tween 10 will be 1 inch. In other words the move commands do not remember to move an object to a particular point on the screen, rather they remember to move the object a certain distance in a certain direction.

So when performing upstream editing remember to check out an object's future and make the proper adjustments to compensate for the changes you've made.

The best tools for doing this kind of editing are the time window and the **see tween** and **replay tween** from the **Time** menu. The **see tween** function is a preference. When set, all tweens are shown on the screen either at their beginning or at their end. The default setting is to show the tweens at their end. By flipping back and forth between the two settings and using the **replay tween** function, it is easy to locate certain movements you may wish to smooth or change. The time window, of course, is essential for jumping to specific areas of the animation; and it can be used to locate the area where the movement takes place, while the other functions provide the ability to pinpoint.

Script Files

The disk files holding the commands for an animation have the extension of **.script** added to them. For example an animation

entitled **cycle demo** appears on the disk directory as **cycle demo.script**.

These are text files. This means they may be edited with a word processor which saves documents in an ASCII text format or any other text processor. You may write your own programs that generate files usable by Aegis Animator. The exact file layout is available from the customer support department at Aegis Development.

If you wish to view a script file:

- Enter the CLI (Command Line Interpreter) portion of the Amiga operating system.
- Place the Animator disk in a disk drive and log onto the main directory. Performing a **dir** (directory) command will list the available files on the disk.
- Choose a file with the **.script** extension (one you have seen in operation is best since the commands will relate to what happens and make more sense).
- At the **1>** prompt enter **type** and then the filename with the extension surrounded by double quotes. For example, a file titled **cycle demo** would be entered like this:

  1>**type "cycle demo.script"**

- The file will appear on the screen. If the file is long it will run by quickly scrolling off the screen. To stop the scrolling so you may read the file, hold the **ctrl** (control) key down and press the **s** key at the same time. An easy way to remember this is **s** for stop. To restart the scrolling press the RETURN key.

Script files list all other files used in the animation at the top. This is handy for quick changes to a file name or for checking that all files needed for an animation are present.

This is an important point to note: *When moving an animation to its own disk you must be sure and move all the associated mask, window, polygon, and picture files as well.*

The script file does not store any of this data, only the name of the file. This allows you a lot of flexibility in being able to change the files from the paint program or polygons from Animator without having to change the actual animation.

Shortcuts for loading Animator

Once you're in the CLI there are a couple of different ways you can boot and run Animator. Each one provides a different shortcut.

At the 1 if you type:

ani This loads the Aegis Animator program without any files open and puts you in the edit mode. This is a normal boot.

ani woozy This version is the same as above except it loads the woozy.script file in ready for editing.

ani -r woozy Loads Animator and plays woozy script. When the animation is complete Animator exits automatically back to the CLI.

ani -r woozy floozy bim Same as above except that all three scripts are replayed before exiting.

ani -m163000 This is a command option that allows you to specify how much memory Animator is to use. The number indicated here is more than Animator uses normally and would not leave enough space for the desktop functions to operate. This is how you can squeeze a little more memory out of the system if you have an extra large animation.

Using Animator with Genlock

One of the most exciting aspects of the Amiga is its interface capability with video. Because the screen output is compatible with VCR, lazer disk, and other video equipment it can interact directly with these devices in a number of different ways. One of the first products to take advantage of this is the Commodore Genlock system.

The Amiga 1300 Genlock system allows users to simultaneously produce video and stereo signals while running programs on their Amiga. The signals can come from a video camera, VCR, lazer disk, video tape, or other computer. Not only can the signals be produced on the machine together with a program but the combined signals can be recorded.

For example, music and video images could be entered into the computer via the Genlock system from a video tape. Imagine at the same time an animation created with Animator playing on the screen. The combined image of the two is then recorded on another video machine. Instant video studio!

The video signal produced by the Genlock system is displayed in the background color (zero). All other colors display over the top of it. The video out portion of Genlock can switch from the Amiga signal to the video signal or produce both at the same time.

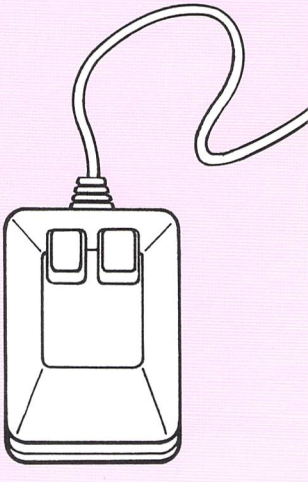
An animated title screen for a video is a good example of this. The animation is first produced on a blank background. When it is complete the video signal from a camera or other input device is entered through Genlock. The animation is begun and then the recording machine is switched on. That's all there is to it. When the animation is complete the recorder is switched off.

Combining the Genlock system with Animator opens an amazing number of creative possibilities. Animations are no longer confined to the amount of memory available and music can be added as well as real visual images.

The Amiga 1300 Genlock system is available through authorized Commodore dealers.

CHAPTER TWELVE

Aegis Animator Reference



Project

Undo	Returns the screen to the way it was before the last command was issued	8
Fast Menu	Hides or displays the fast menu window	11,93
New Script	Erases the current script	00
Storage	Displays the storage requestor	76
Storyboard	Transfers program control to the storyboard	9,63
Color	Displays the color requestor	55
Time	Displays the time requestor	53
Status	Displays in the menu bar the remaining available memory	65
Exit	Exits the Animator program	

Create

Line	Creates an object that is a single line	24
Polygon		
Filled	Creates a solid free form shape	25
Outline	Creates an outline free form shape	

Circle

Filled Creates a solid circle made of sixteen points 25

Outline Creates an outline circle as above 00

Star

Filled Creates a solid five pointed star 26

Outline Creates an outline five pointed star 00

Block Creates a four sided rastor object It is in effect the same as a mask 26

Clone Creates an exact copy of an object or objects 27

Destroy Removes an objects present and all of it's future 27

Move

Move

Sideways Drags the current selection about the screen 30

In Moves the current selection **inward** through successive planes The further the mouse is moved the further the selection will travel The fast menu selection of this command has an added feature of **perspective on** or **off** The default is **on** which means the selected objects will get smaller as you move them **in** because they are moving away from you With the perspective turned **off** the selected objects move **inward** but do not change size 31

Out This is the same as the **in** command except the selected objects move **outward** 31

Rotate

In Plane Moves the current selection about an indicated center without changing the plane the selection inhabits 33

Around X Moves the current selection into the **x** plane around an indicated horizontal center 34

Around Y	Moves the current selection into the **y** plane around an indicated vertical center	35
Size	Shrinks or enlarges the current selection toward an indicated center	37
Path	Plots a path for the current selection to follow	38
Change Color	Changes the color of the current selection to the current color at the beginning of the tween	39
Change Type	Changes the type of the current selection Types are filled (solid), outline, and line These are the basic shapes used in Animator	40
Morph		
Loop	Changes the shape or position of the current selection by moving the points that make up the objects	41
Hook	Changes the shape or position of the current selection by moving existing points or adding and deleting the points that make up the object	42

Select

Point	Makes the current selection active on one point at a time	47
Points	Makes the current selection active on multiple points at the same time The points can be located on different objects	47
Segment	Makes the current selection active on a portion of a single object	48
Polygon	Makes the current selection any one object	48
Polygons	Makes the current selection any number of multiple objects	49
All	Selects all objects currently in the animation	49

Time

Next Tween Advances the current position to the next tween If there are no more tweens it creates one and makes it the current tween 52

Replay Tween Plays back the animation in the current tween ... 00

Replay All Plays back the entire animation

Play Loop Plays the entire animation and loops back to the beginning It will continue to repeat until interrupted 00

Ghost Mode Displays all objects in wire graphics This is useful for getting at objects behind others, or for just seeing what's going on 9

See Tween

At End When in edit mode all tweens will be displayed at the end (objects will appear where they will be when the tween completes) 9,83

At Begin This preference is the same as **at end** except the tween will display at its beginning 83

Color

This is the current color palette The current color is highlighted A new color may be chosen from here

Storyboard

Project

Go Into Enters the edit mode in the selected window ... 64

Speed Displays the time requestor 65

About Ani Displays information about the creation of this program 00

Status Displays in the menu bar the remaining available memory 65

Exit Exits the Animator program 65

Edit

Splice Copies the current tween and all future tweens into another window 66

Cut Cuts the current tween and all future tweens into another window 66

Delete Deletes all animation from the indicated window 67

Activate Works the same as **play loop** in the indicated window 67

The Fast Menu

The fast menu is an iconic method of performing the same tasks available under the regular menus. Each symbol in the menu represents a command from the menu bar. The only function not represented in the menu bar is the twisting shape next to the **undo** in the upper right corner of the menu window. You might think of this as a function key. Some of the commands have more options than can be displayed in the window at the same time. For example, the polygon create symbol (pencil like shape in lower left corner) can be used to create any of the objects listed under the **Create** menu option. First select the function key symbol in the upper right corner, then select the polygon create symbol. A window will appear with available options. Set the selections to your choice and press the Menu button when you are through.

Figure 12.1 — The Fast Menu.

Glossary

AmigaDOS—The Amiga disk operating system.

Background—A special rastor file that is behind all other objects. Stored on disk as a **.pic** file.

Cancel—To back out of an operation without performing any action. Usually performed by pressing the Menu (right) mouse button.

Cel—short for cellophane. A type of animation popularized in the '50's when it became too expensive to have artists draw each frame. In cel animation small pieces of the animation (ie. a face, hand, or arm) are painted on a transparent piece of cellophane. The cells are moved independently of each other from frame to frame, but not usually redrawn. Sideways and up and down motion are generally possible with cel animation. Cel is also used to refer to **.win** file (window) objects.

choose—To pick a menu item. You normally choose menu items with the Menu button.

click—To press and release a mouse button.

close—To remove a window, or requestor from the display. To put away a tool, animation, or file.

color—In this manual it usually refers to one of the slots in the Color Map.

Color Bar—The set of colors available for use displayed as a bar. These appear in the Fast Menu and the Color Requestor. Also known as the Color Palette.

color cycling—A special effect that allows you to change a series of colors at regular intervals. You can use color cycling to create an illusion of motion.

Color Map—A table used to translate from the numerical representation of data in the computer to the representation as colors on the screen. The Color Map can be thought of as a series of ink wells numbered from left to right. The type of ink in the wells can change, but a picture that calls for ink 0 will

use the ink from the left-most bottle. Individual Color Maps can be stored on disk as a **.col** file (color).

Color Requestor—The requestor you use to change the colors in the Color Palette. It also contains the commands for cycling and fading a series of colors. Reached by selecting **color** from the **Project** menu.

cursor—The icon that moves on the screen when you move the mouse. You use it to select menu items, work gadgets, create objects etc. The cursor changes shape depending on what tool has been selected.

cut—Refers to an editing process in the storyboard.

disk—A magnetic medium for storing and retrieving information. The little plastic things with the metal strip that opens and closes.

end color—One of two colors at either end of a range.

filled—Refers to objects displayed in an animation. If they are filled they appear solid.

highlight—To display something in a way that distinguishes it. Normally, something is highlighted to indicate it is selected.

icon—A visual representation of a tool or function.

Mask—A rastor object with only a single color and a transparent background. Masks are stored on disk in **.msk** files.

Menu Bar—A strip at the top of the screen containing menu titles and instructions for the selected function.

Menu button—The right hand button on the mouse.

Object—Any thing such as a line, circle or polygon that can be isolated and moved. This includes rastors such as masks and windows.

open—1. To make a tool available for use. 2. To retrieve a file stored on disk. 3. To display a window or requestor.

Pendown—Pressing down on the Selection (left) mouse button.

Penup—Releasing pressure on the Selection (left) mouse button.

Pixel—One of the individual points that make an image on the screen.

Polygon—An object defined by a number of points and straight lines between these points. Polygons can be stored by themselves on disk in a **.poly** file.

range—A set of intermediate colors whose values are between two end colors.

Raster—An object that conceptually corresponds to a sheet of graph paper with some of the squares colored in. Also known as a bit image. Typically such an object can be drawn faster than a polygon, and is necessary for images with a great deal of detail. Rasters are stored on disk as **.win** or **.msk** files. *See also Cel, Mask, Window.*

requestor—A bounded region on a screen used to give information to a tool or animation. It appears as a window when you select certain tools. Most requestors have gadgets that control different aspects of the animation. When you exit Animator a requestor appears asking if you really want to leave. There are two buttons **yes** and **no** to answer the request. The main requestors have close boxes in them located in the extreme upper left corner. By clicking here the requestor will go away.

save—To copy something to a file on disk.

Screen—The display area of the monitor, or the rastor that holds the background and all other objects drawn on top of the background.

Script— An animation. The information that is stored on disk to re-create an animation. Stored on disk as a **.script** file. These files are in text format and may be looked at or edited using a text editing program.

select—To pick an icon, object, or gadget by pointing to it with the cursor and pressing the Selection button.

Slider—A gadget you use to pick within a range by dragging a marker along a line.

Stencil—Another word for Mask. *See Mask.*

Storyboard—A place for editing scripts. The storyboard has nine slots each one capable of holding a different script. The editing that takes place here is like film editing (splicing, cutting, etc.).

Tween—A segment of time. A portion of a script.

Upstream Editing—Changing an object in a script when that object already has movements and changes defined in the future from the point of the current edit.

window—1. Program window, the current window Animator is running in on the Amiga. 2. Rastor window, an image captured from a painting and moved about the screen, *see Rastor.* 3. Requestor window, a requestor area on the screen, see requestor. 4. Storyboard window, one of nine slots each of which can hold a separate animation.

Index

A

about ani 10,91
action 62,76,76
activate 10,66,67,92
advance next tween 11
Aegis Images 2,24,70
all 46,49,90
ani 6,85
ani -m 85
ani -r 85
ani-panicx.script 68
around X 32,34,35,89
around Y 32,35,89
ASCII 84
at begin 91
at end 91

B

backdrop 24,58,70,76,81
bit map images 71
block 26,89

C

camera icon 11,57,59
cd 71,76
cel animation 14
cellophane 14
change color 39,40,55,58,90
change type 40,90
circles 25,89
CLI 6,8,10,84
clone 8,27,89
close box 7
col 61,76
color 8,9,88,91
color bar 11,55,57,58,61
color bundles 61,76
color control 55
color control requestor 11
color menu 16,39,40,55,58
color palette 62
color requestor 56
color window 58
create 8,15,24,25,26,27,28,
40,65,80,88,92
current color 56
cursor hot spot 16
cut 10,65,66,92
cycle 58,59,70

D

delete 10,65,67,92
destroy 8,27,28,89
destroy icon 67
dir 84
do fills 65

E

edit 8,10,64,65,66,
67,88,92

F

fade 59,60
fast menu 7,10,11,88,92,93
filled 8,15,25,26,88,89
function key 92

G

Genlock 86
ghost mode 9,65,68,91
global speed 54
go into 64,91

H

hook 41,90
hot spot 16
hue 57

I

IFF 70
in 32,89
In and Out 31
in plane 32,33,34,89
into 10,64
Introduction to Amiga 2,6

J

jump to tween 55

K

kill 76

L

light 57
line 88
line create 24
lines 24
load 62,70,71,73,74,76
local speed 54,55
loop 41,90
low resolution 80

M

main screen 7
mask 68,70,73,74,76,81,82
memory panic 68
metamorphic 30
metamorphic animation 14
morph 41,43,90
morphing 8
move 8,18,19,20,21,24,26, 27,31,32,33,34,36,38, 39,40,41,43,46,47, 48,55,58,71,83,89
move in 72,82
move out 72
move sideways 16,27,33,42,72
msk 73,74,76

N

new script 8,15,67,88
next tween 11,17,19,20, 21,52,82,91

O

out 32,89
outline 8,9,25,26,88,89

P

palette 76
palette icon 10
path 38,39,72,90
pic 70,71,76
picture 81
plane 31,32,33,34,81
play loop 20,67,91
point 46,47,90
points 46,47,90
poly 76
polygon 8,15,16,37, 46,48,88,90
polygon filled 80
polygons 9,25,26,46,49, 73,76,81,90
project 7,8,10,88,91
project menu 24

R

range 60
range selector 57,59,60,61
raster 26,71,73,82
replay all 19,21,22,91
replay tween 17,83,91
RGB 56
rotate 37,46,47,49,89
rotate around x 19,34
rotate around y 20,36
rotate in plane 18,30,33,43
rotation 32

S

saturation 57
save 62,72,74,76,81
save window 71
scissors 66
script 76
script files 83
see tween 83,91
segment 9,46,48,90
select 8,24,46,47,90
sideways 30,31,89
size 8,21,37,38,89
spectrum 61
speed 26,65,68,82,91
splice 10,64,65,66,92
star 26,89
start 61
status 8,64,65,88,91
storage 61,62,71,72,74,76,88
storage requestor 70
storage window 76,78,81
storyboard 2,8,9,64, 66,81,88,91
suffix 61,76

T

text files 84
text scrolling 81
time 8,9,11,17,19,20, 21,22,52,65, 66,68,83,88,91
time requestor 54
title 76
to filled 40
to line 40
to outline 40
tween 8,14,15,28,30, 31,33,34,36,37, 38,40,41,50,52,58
tweening 14,52
type 84

U

undo 8,24,57,64,88,92
upstream editing 83

W

win 72,76
window 68,70,71, 72,76,81,82
wipe 57,70
workbench 6,8,10,65

X

X axis 34,35

Y

Y axis 36

Colophon

This book was input on a 512K Macintosh *using* Microsoft Word.

The text files were transferred to an IBM AT *using* Aegis Mac Mail *and* Mycroft Labs MITE. *Files were then coded and massaged for typesetting and transferred to a* Varityper EPICS *system by* Dynatype Telecom *of Glendale, California.*

Cover design by Chris Savage of Peter Green Design, *Toluca Lake, California.*

Interior design by Dave Barrett and Paul H. Smith.

Printing on a Two-Color Heidelberg *by* AVL Press *of Burbank, California.*

Book Text:
ITC Garamond Ultra, Bold *and* Light
Helvetica Bold

Book Title and Chapter Pages:
Digitally condensed Helvetica Extra Bold

Call-Outs:
Brush

Papers:
Interior — 60lb. White Offset
Cover — 100lb. Coated

INSIDE AEGIS ANIMATOR

Affordable full color animation has come to the microcomputer world at last with the introduction of Aegis Animator. This guide designed for both the beginner and experienced user takes you on a step by step tour of this remarkable program. From the concepts of metamorphic animation to shortcuts and technique, it details all you need to know to produce quality animation.

Topics covered include:

- Tweening
- Metamorphic Animation
- Shapes, movement, and color
- The Storyboard
- Using images created outside Animator
- Animation technique

$14.95

ISBN: 0-915595-10-9

PaperDisk Publishing
Los Angeles, CA